Paweł Łupkowski

TEST TURINGA

PERSPEKTYWA SĘDZIEGO

UMYSŁ

PRACE Z FILOZOFII I KOGNITYWISTYKI

Recenzent: prof. dr hab. Adam Grobler

Publikacja dofinansowana przez Instytut Psychologii
Uniwersytetu im. Adama Mickiewicza w Poznaniu

Projekt okładki: Marta i Zdzisław Kwiatkowscy
Redaktor: Elżbieta Piechorowska
Redaktor techniczny: Elżbieta Rygielska
Łamanie komputerowe: Paweł Łupkowski

ISBN 978-83-232-2208-8

WYDAWNICTWO NAUKOWE
UNIWERSYTETU IM. ADAMA MICKIEWICZA W POZNANIU
61-701 Poznań, ul. Fredry 10
www.press.amu.edu.pl
Sekretariat: tel. 61 829 46 46, faks 61 829 46 47, e-mail: wydnauk@amu.edu.pl
Dział sprzedaży: tel. 61 829 46 40, e-mail: press@amu.edu.pl
Wydanie I. Ark. wyd. 9,25. Ark. druk. 7,5
Druk i oprawa: QUICK-DRUK s.c., ŁÓDŹ, ul. ŁĄKOWA 11

Spis treści

Wstęp

> Pomimo rozlicznych prób zdyskredytowania testu Turinga okazał się on niesłychanie odporną propozycją na gruncie filozofii umysłu i filozoficznych dyskusji o sztucznej inteligencji. W momencie, w którym filozoficzni przeciwnicy testu Turinga obwieszczają jego zupełne unicestwienie, on — niczym feniks z popiołów — odradza się wraz z nowymi obrońcami przedstawiającymi nowe tezy przemawiające na jego korzyść[1].
>
> L. J. Crockett (1994) *The Turing Test and the Frame Problem. AI's Mistaken Understanding of Intelligence*

Mianem testu Turinga (TT) określa się propozycję gry przedstawioną przez Alana Turinga w jego znanym artykule „Computing Machinery and Intelligence", który ukazał się w czasopiśmie *Mind* w roku 1950. Punktem wyjścia w tym tekście jest pytanie o to, czy maszyny mogą myśleć? Turingowi pytanie to wydaje się zbyt wieloznaczne, by zasługiwało na dyskusję, więc proponuje zastąpienie go innym, na które prawdopodobnie łatwiej będzie uzyskać odpowiedź: *czy w przypadku pewnej gry maszyna poradzi sobie równie dobrze jak człowiek?* Konstrukcję i zasady owej gry zaczerpnął Turing z gry towarzyskiej, nazywanej grą w naśladownictwo (*imitation game*). Biorą w niej udział trzy osoby: mężczyzna (A), kobieta (B) i pytający (C) (dowolnej płci, nazywany również sędzią). Mężczyzna i kobieta przebywają w osobnych pokojach, oddzieleni od siebie i od pytającego. Oczywiście gracze nie mogą się widzieć ani słyszeć, czy też pisać do siebie pismem odręcznym (mogą porozumiewać się np. dzięki gońcowi). Pytający może zadawać pytania osobom w pokojach, które to osoby zna jako X i Y. Jego zadaniem jest określenie (wyłącznie na podstawie uzyskanych odpowiedzi), w którym pokoju znajduje się kobieta, a w którym mężczyzna. Modyfikacja wprowadzona przez Turinga polega na zastąpieniu jednej z osób maszyną. Pisze on: „Zadajmy teraz pytanie »co

[1] Jeśli w bibliografii pracy nie zaznaczono inaczej, tłumaczenia dokonał P. Łupkowski.

stanie się, kiedy maszyna weźmie udział w tej grze?« Czy pytający dokona błędnej identyfikacji równie często, jak w przypadku, gdy w grze biorą udział kobieta i mężczyzna?" [Turing 1950, s. 434]. Oczywiście zadaniem pytającego nie jest już odgadnięcie płci jednego z graczy, ale rozpoznanie — podobnie jak w przypadku gry w naśladownictwo jedynie na podstawie udzielonych odpowiedzi — który z nich jest człowiekiem, a który maszyną. Kryterium bycia istotą inteligentną zostaje tym samym sprowadzone do kryterium powodzenia w tak zaprojektowanej grze. Jeżeli pytający zidentyfikuje maszynę jako człowieka, uzyskamy podstawy do stwierdzenia, że owa maszyna jest inteligentna.

Test Turinga zbliża się już do swoich sześćdziesiątych urodzin ale z całą pewnością — wbrew licznym głosom krytyków — nie stanowi dziś zagadnienia przebrzmiałego, o którym powiedziano już wszystko, co było do powiedzenia. Należy jednak uczciwie przyznać, że powiedziano bardzo wiele. Propozycja A. M. Turinga, przedstawiona w „Computing Machinery and Intelligence" stanowi bowiem źródło inspiracji w wielu dyscyplinach. Zagadnienie testu Turinga poruszane jest w pozycjach zaliczanych do psychologii (por. np. [Nęcka 2005], [Watt 1996]), kognitywistyki (por. np. [Konar 2000], [Harnish 2002], [Casacuberta 2007]), informatyki (por. np. [Tanimoto 1987], [Luger, Stubblefield 1998], [Thro 1994]) czy filozofii (por. np. [Searle 1995], [Hetmański 2000]). Test Turinga znalazł swoje miejsce również poza nauką — w literaturze popularnej (por. np. *Neuromancer* Williama Gibsona) a nawet w muzyce (opera *The Turing Test* autorstwa Juliana Wagstaffa[2]).

Dyskusje prowadzone wokół testu Turinga nie zamykają się w ramach czysto teoretycznych, akademickich rozważań, ale wkraczają również na obszar zagadnień praktycznych. Test Turinga generuje wiele problemów, które umieszczają go w szerszej perspektywie rozważań dotyczących zagadnienia sztucznej inteligencji:

— *Czy rzeczywiście istnieje jakiś związek pomiędzy testem Turinga a posiadaniem inteligencji? Czy test Turinga jest dobrze zaprojektowany dla celów, którym ma służyć?* Na te pytania udzielane są rozmaite odpowiedzi, od stanowisk krytykujących test Turinga, poprzez próby jego wzmocnienia lub osłabienia, aż po głosy broniące propozycji Turinga.

— *Czy test Turinga dostarcza definicji inteligencji jako takiej?* Przy próbie odpowiedzi na to pytanie badacze starają się wskazywać na rodzaj uzasadnienia tezy o inteligencji maszyn, jaki oferuje test Turinga. Czy mamy tu do czynienia z operacyjną definicją inteligencji, czy też może intencje Turinga szły w zupełnie inną stronę?

— *Czy wybór celu badań wyznaczonego przez test Turinga jest korzystny dla rozwoju sztucznej inteligencji jako dyscypliny?* Pytanie to zazwyczaj pojawia się w kontekście krytyki testu Turinga jako pewnego celu wyznaczonego dla badaczy sztucznej inteligencji — „zbuduj maszynę, która pod względem zachowań językowych będzie nieodróżnialna od

---

[2] Opera miała premierę 15 sierpnia 2007 roku. Można jej posłuchać na http://www.julianwagstaff.com/ttt/music.html.

człowieka". Oczywiście owa krytyka zakorzeniona jest głęboko w negatywnych odpowiedziach na dwa wcześniejsze pytania.

— *Czy maszyna, która zda test Turinga powinna być traktowana jak osoba ludzka?* Odpowiedź na ten problem związana jest z całą — długą już — tradycją rozważań etycznych skupionych wokół zagadnienia statusu maszyn myślących (por. np. [Jonas 1996], [Kiepas 1992], [Łupkowski 2005*b*]).

— *Jak zaprojektować program, który zda test Turinga?* Jest to oczywiście pytanie, które zadają sobie informatycy, stawiający sobie za cel stworzenie programów zdolnych do porozumiewania się z użytkownikiem przy użyciu języka naturalnego.

— *Czy idea testu Turinga może pomóc w praktycznym odróżnianiu ludzi od maszyn?* Na pytanie to — jak się wydaje, jedyne spośród wymienionych — można już dziś udzielić odpowiedzi twierdzącej, co zrobili badacze zajmujący się automatycznymi systemami autoryzacji użytkownika (por. rozdział 4).

Jeżeli spróbujemy prześledzić liczne próby udzielenia odpowiedzi na przedstawione pytania, łatwo daje się zauważyć, że duża część prowadzonych dyskusji wydaje się mocno odbiegać od tego, co można by nazwać oryginalną propozycją A. M. Turinga. Częste są próby rekonstruowania testu Turinga jedynie na podstawie „Computing Machinery...", z zupełnym pominięciem późniejszych źródeł dotyczących testu. Często autorzy — zwłaszcza stanowisk krytycznych — nie zaprzątają sobie głowy szczegółową rekonstrukcją testu Turinga, bazując jedynie na ogólnej jego idei (czy może lepiej, obiegowym wyobrażeniu o nim). Taki stan rzeczy stał się dla mnie motywacją dla próby możliwie kompleksowego odtworzenia oryginalnych założeń testu Turinga w oparciu o jak najbogatsze źródła autorstwa samego Turinga. Dzięki temu możliwe jest uznanie pewnych dyskusji toczonych wokół zagadnienia testu Turinga za bezcelowe, a nawet bezpodstawne. Można w tym kontekście przywołać przykład próby tak zwanego „literalnego" odczytywania „Computing Machinery...", którego zwolennicy uważają, że w rzeczywistości Turing zaproponował test płci (por. rozdział 1.4). Szczegółowa rekonstrukcja oryginalnych założeń testu Turinga oraz dyskusja z pewnymi nieporozumieniami spotykanymi w literaturze przedmiotu stanowią tematykę pierwszego rozdziału niniejszej książki.

Rozdział drugi poświęcam omówieniu i skomentowaniu sporów oraz dyskusji prowadzonych wokół testu Turinga. Podejmuję w nim próbę sklasyfikowania argumentów krytycznych wysuwanych wobec testu Turinga, korzystając przy tym z zaproponowanego przez R. Frencha rozróżnienia na tezę filozoficzną i tezę pragmatyczną testu Turinga. W rozdziale drugim rozważam również konsekwencje pewnych wyników uzyskanych na gruncie logiki pytań dla zagadnienia adekwatności testu Turinga, jako pewnego kryterium badania obecności inteligencji w systemach sztucznych.

Rozdział trzeci stanowi spojrzenie na test Turinga z perspektywy sędziego (pytającego). Podejmuję w nim próbę formalnej analizy testu z wykorzystaniem narzędzi opracowanych w ramach inferencyjnej logiki pytań. Dzięki tej analizie mam nadzieję lepiej zrozumieć główne założenia dotyczące testu Tu-

ringa i uwidocznić rolę sędziego w przebiegu testu. Zastanowię się również nad istnieniem optymalnej strategii dla sędziego, która umożliwiałaby dokonanie trafnej identyfikacji gracza.

W czwartym rozdziale niniejszej książki przedstawię najciekawsze, moim zdaniem, koncepcje teoretyczne oraz rozwiązania praktyczne, które w sposób bezpośredni nawiązują do idei zawartych w teście Turinga. Omówione zostaną propozycje testów alternatywnych względem TT: odwrócony test Turinga (*Inverted Turing Test*) autorstwa S. Watta, *Minimum Intelligence Signal Test* autorstwa Ch. McKinstry'ego oraz test lady Lovelace (*Lovelace Test*) zaproponowany przez S. Bringsjorda. Przedstawię również praktyczną realizację idei testu Turinga, którą stanowi pewna klasa systemów automatycznej autoryzacji użytkownika powszechnie określana mianem CAPTCHA.

Książkę kończy dodatek zawierający krótką biografię Alana Mathisona Turinga.

Podziękowania

Książka ta jest rozszerzoną i poprawioną wersją pracy doktorskiej obronionej w 2009 r. w Instytucie Filozofii Uniwersytetu im. Marii Curie-Skłodowskiej w Lublinie.

Bardzo serdecznie dziękuję recenzentom tej pracy — prof. Adamowi Groblerowi i prof. Markowi Hetmańskiemu — a także dr. Mariuszowi Urbańskiemu za wszelkie uwagi i sugestie, które znacząco wpłynęły na jej ostateczny kształt. Szczególnie pragnę podziękować mojemu promotorowi prof. Andrzejowi Wiśniewskiemu za cierpliwość i wyrozumiałość oraz za cenne uwagi i czas poświęcony na długie dyskusje, dzięki którym powstała niniejsza praca.

Rozdział 1

Test Turinga

1.1. Historia idei testu Turinga

Alan Turing nie był pierwszym, który zadawał sobie pytanie o to, czy maszyny mogą myśleć. Pojawiło się ono, w naturalny sposób, wraz z powstaniem maszyn, których poziom skomplikowania umożliwiał imitację pewnych zachowań istot żywych. Badacze zajmujący się zagadnieniem testu Turinga wskazują na Kartezjusza jako na tego, który pierwszy zdał sobie sprawę z problemu, jaki niesie ze sobą powstanie wyrafinowanych maszyn (por. [Gunderson 1964], [Copeland 2000], [Sterrett 2000], [Erion 2001], [Shieber 2004], [Bringsjord 2009], [Chomsky 2009]). Kartezjusz, zafascynowany współczesnymi sobie automatami, porównywał do nich zwierzęta i ciało ludzkie. W części V *Rozprawy o metodzie* pisze on:

> Nie wyda się to zgoła dziwne tym, którzy wiedząc, ile rozmaitych automatów, czyli poruszających się maszyn, przemyślność ludzka umie wykonać używając niewielu jeno części w porównaniu do wielkiej ilości kości, mięśni, nerwów, tętnic, żył i wszystkich innych składników, jakie są w ciele każdego zwierzęcia, uważać będą to ciało za maszynę, która, jako uczyniona rękami Boga, jest bez porównania lepiej obmyślona i zawiera w sobie ruchy bardziej godne podziwienia niż jakakolwiek stworzona przez człowieka [Kartezjusz 1637/1994, s. 42].

Zarówno zwierzęta, jak i ludzkie ciało są, zdaniem Kartezjusza, automatami. Tym jednak, co odróżnia człowieka od zwierząt, jest posiadanie przez niego duszy (którą obdarował człowieka Bóg).

Dalej Kartezjusz pisze:

> Zatrzymałem się też tu umyślnie dla wykazania, że gdyby istniały takie maszyny, które miałyby narządy i zewnętrzną postać małpy lub innego jakiego bezrozumnego zwierzęcia, nie mielibyśmy sposobu rozpoznać, że nie są one we wszystkim tej samej natury co owe zwierzęta [...] [Kartezjusz 1637/1994, s. 42].

Z odmienną sytuacją mamy jednak do czynienia w przypadku automatów, które miałyby naśladować ludzi:

> [...] podczas gdyby istniały maszyny, podobne do naszych ciał i naśladujące nasze uczynki na tyle, ile byłoby to w zasadzie samej możliwe, to mielibyśmy zawsze dwa bardzo pewne sposoby rozpoznania, że jeszcze dzięki temu nie byłyby one prawdziwymi

ludźmi. Pierwszy ten, iż nigdy *nie mogłyby używać słów ani innych znaków składając je w ten sposób, jak my czynimy dla oznajmienia innym naszych myśli.* Można bowiem pojąć, iżby maszyna tak była zrobiona, że wymawia jakieś słowa, a nawet wymawia ich kilka w związku z działaniem fizycznym powodującym pewne zmiany w jej przyrządach: jak to, kiedy się ją dotknie w jakimś miejscu, aby spytała czego sobie od niej życzymy; w innym, aby krzyczała, że ją boli, i tym podobne; ale niemożliwe jest, aby składała rozmaicie słowa, odpowiadając do sensu na wszystko, co się powie w jej obecności, jak to ludzie bodaj najbardziej tępi mogą czynić. Drugi sposób jest ten: *choćby nawet maszyny takie czyniły wiele rzeczy równie dobrze lub może lepiej niż którykolwiek z nas, nie robiłyby niezawodnie wielu innych, i przez to można by odkryć, iż nie działają dzięki świadomości, lecz jedynie dzięki rozmieszczeniu swoich przyrządów.* Podczas bowiem gdy rozum jest to instrument wszechstronny, który może służyć we wszelkiego rodzaju przypadkach, te przyrządy potrzebują pewnego szczególnego ustawienia dla każdej poszczególnej czynności; skąd pochodzi, że zasadniczo niemożliwe jest, aby w maszynie była dostateczna ich rozmaitość, by mogły ją wprawić w działanie we wszystkich okolicznościach życia w taki sam sposób, w jaki nasz rozum powoduje nasze działanie [Kartezjusz 1637/1994, s. 42–43]; wyróżnienia P.Ł.

Powyższy fragment *Rozprawy o metodzie* zawiera propozycję kryteriów, odróżniających automaty (w tym również zwierzęta) od ludzi. G. Erion [2001] nazywa tę propozycję kartezjańskim testem automatyzmu (*The Cartesian Test for Automatism*). Jak łatwo zauważyć, składa się on z dwóch elementów: testu językowego (*language test*) i testu działań (*action test*) (por. [Gunderson 1964, s. 198]).

Test językowy opiera się na założeniu, że automat nigdy nie będzie w stanie opanować ludzkiej mowy, ponieważ jest ona zbyt wyrafinowana i złożona. Nawet jeśli moglibyśmy sobie wyobrazić, że udałoby się skonstruować automat, który potrafiłby naśladować pewne elementy ludzkiej mowy, to i tak z łatwością można by odróżnić jego wypowiedzi od wypowiedzi człowieka. Po pierwsze, nie jest możliwe, żeby maszyna poradziła sobie z wielością kontekstów dowolnego, realistycznego dialogu między ludźmi. Po drugie, pozbawione ludzkiego umysłu automaty nie mogą w rozmowie wyrażać swoich własnych myśli, tak jak robią to ludzie. Po trzecie wreszcie, automatom nie przysługuje „rzeczywista mowa", jak określa Kartezjusz możliwość wyrażania abstrakcyjnych myśli (np. o matematyce). Automaty mogą więc jedynie naśladować ludzką mowę (czy też posługiwać się pewnymi jej elementami), ale nigdy nie będzie możliwe, żeby takie naśladownictwo dorównało poziomowi oryginału.

Test działań opiera się na założeniu, że ludzie, podejmując działanie, postępują wedle zasad, o których mogą rozumować, mogą je oceniać i zmieniać w zależności od swoich planów. Automaty — co Kartezjusz pokazuje na przykładzie zwierząt — mogą jedynie trzymać się wyznaczonych reguł i zadanych strategii. Innymi słowy, test działań opiera się na obserwacji elastyczności zachowań w zmieniających się warunkach i środowisku. Automat będzie zawsze radził sobie albo zbyt dobrze (np. jeżeli chodzi o zadania arytmetyczne), albo zbyt słabo, aby uznano go za człowieka (oczywiście bardziej prawdopodobna jest ta druga klasa przypadków).

Z propozycją Kartezjusza wiąże się ważne pytanie: czy automat powinien przejść oba etapy testu pozytywnie, czy też wystarczy powodzenie w jednym z nich, aby uznać, że ów automat jest inteligentny. Nasuwającym się rozwiązaniem jest uznanie obu etapów testu za w pewnym sensie równie istotne.

Zarówno w teście językowym, jak i w teście działań badane są zachowania systemu (czy to pod względem jego wypowiedzi, czy też innych behawioralnych kryteriów). Ciekawą interpretację proponuje w tym kontekście G. Erion. Jego zdaniem na test działań powinniśmy patrzeć jako na test posiadania własności, którą określa on mianem zdrowego rozsądku (*common sense*). Pod tym pojęciem Erion rozumie:

> [...] elementarną wiedzę o zdroworozsądkowo pojmowanej rzeczywistości, która umożliwia nam przetrwanie i ułatwia funkcjonowanie w codziennym życiu. Powszechnie żywione przekonania dotyczące zdroworozsądkowo pojmowanego świata stanowią najbardziej znaczący komponent tej elementarnej wiedzy [Erion 2001, s. 33].

Jeżeli zgodzimy się z takim stanowiskiem, to test językowy staje się w pewnym sensie częścią testu działań.

> Automaty różnią się od ludzi w dwóch kwestiach. Po pierwsze, automaty nie mogą używać języka. Po drugie, automaty nie posiadają zdrowego rozsądku (*common sense*), na który składa się nie tylko wiedza dotycząca używania języka, ale również wiedza o tym, jak wykonywać pewne zadania czy też odpowiadać na zadane pytania, na które nawet nieco nierozgarnięty dorosły człowiek jest w stanie odpowiedzieć. Mówiąc inaczej, automatom brak zdrowego rozsądku (*common sense*), a tym samym tej jego części, która związana jest z kompetencją językową [Erion 2001, s. 36].

Kartezjański test automatyzmu wyprzedza tym samym rozważania badaczy współcześnie zajmujących się testem Turinga. Daje się zauważyć wyraźną analogię między kartezjańskim testem działań a propozycją poszerzenia testu Turinga autorstwa S. Harnada (por. rozdział 2). Podobnie jest, jeśli chodzi o kartezjański test językowy i — oparty na tzw. pytaniach subkognitywnych — argument R. Frencha wymierzony w test Turinga (por. rozdział 2). Można również znaleźć podobieństwo pomiędzy założeniami leżącymi u podstaw kartezjańskiego testu działań a tzw. zarzutem lady Lovelace rozpatrywanym przez A. M. Turinga w „Computing Machinery..." (por. rozdział 2) oraz testem lady Lovelace zaproponowanym przez S. Bringsjorda (por. rozdział 4).

Sam Kartezjusz udziela negatywnej odpowiedzi na pytanie, czy — w sensie zaproponowanych kryteriów — maszyny mogłyby myśleć. Warto jednak wspomnieć o myślicielu, którego poglądy byłyby z całą pewnością bliższe Turingowi — o J. O. La Mettriem. W *Człowieku maszynie* pisze on bowiem:

> Można być maszyną, a zarazem czuć, myśleć, odróżniać dobro od zła równie dobrze jak barwę niebieską od żółtej — innymi słowy, można posiadać wrodzoną inteligencję i moralność, pozostając jednocześnie tylko zwierzęciem: tkwi w tym nie większa sprzeczność niż w fakcie, że można być małpą albo papugą i doznawać rozkoszy [...]. Moim zdaniem, myślenie jest tak dalece nieodłączne od materii zorganizowanej, że wydaje się ono jej właściwością w równym stopniu jak elektryczność, zdolność ruchu, nieprzenikliwość, rozciągłość itd. [La Mettrie 1748/1984, s. 84].

> Wnioskujemy zatem śmiało, że człowiek jest maszyną i że w całym wszechświecie istnieje tylko jedna substancja, występująca w rozmaitych postaciach [La Mettrie 1748/1984, s. 91].

Stanowisko La Mettriego motywowane było odkryciami dotyczącymi świata zwierząt, ale przede wszystkim zachwytem nad możliwościami ówczesnej techniki. Wydaje się, że propozycja testu Turinga i optymizm jego autora co

do stworzenia myślących maszyn wyrastają z takich samych pobudek. Wniosek taki nasuwa się, gdy prześledzimy proces, jaki doprowadził Turinga do sformułowania propozycji znanej dziś jako test Turinga. Jednym z problemów, które szczególnie zajmowały Turinga, było zagadnienie obliczalności. Badania nad tym zagadnieniem doprowadziły go do sformułowania pojęcia maszyny Turinga (MT). Owa abstrakcyjna maszyna jest ogólnym modelem obliczania, o którym mówi się również, że jest abstrakcyjnym modelem komputera (por. [Aho et al. 2003], [Hopcroft, Ullman 2003], [Papadimitriou 2002]). Jednym z rodzajów maszyn Turinga są tak zwane uniwersalne maszyny Turinga (UMT), które charakteryzują się tym, że mogą naśladować dowolną inną MT. Turing pisze:

> W celu umożliwienia naszemu komputerowi imitowania danej maszyny jedyne, co musimy zrobić, to tak zaprogramować ów komputer aby *obliczył, co rozważana przez nas maszyna zrobiłaby w danych okolicznościach* (w szczególności, jakie odpowiedzi by wydrukowała). Odpowiednio zaprogramowany komputer zdolny będzie do udzielania takich samych odpowiedzi [Turing 1951*a*, s. 2]; wyróżnienie P.Ł.

Imitacja jednej maszyny przez drugą polega więc nie tyle na odwzorowywaniu jej wewnętrznej budowy, ile na naśladowaniu jej funkcji. Jeśli dodamy do tego fascynację Turinga możliwościami UMT i jego przekonanie o tym, że stany UMT można porównywać do stanów umysłu, a budowanie fizycznego komputera do budowania mózgu (por. [Hodges 1998, s. 48], [Hodges 1992, s. 290]), poszerzenie zakresu imitacji wydaje się naturalnym krokiem:

> Jeśli teraz pewną maszynę mielibyśmy opisać jako mózg, jedyne co musielibyśmy zrobić to tak zaprogramować ów komputer cyfrowy aby ten mózg imitował. Jeżeli przyjmiemy, że rzeczywiste mózgi [...] są w pewnym sensie maszynami, to następstwem tego będzie uznanie, że — odpowiednio zaprogramowany komputer cyfrowy — *będzie się zachowywał jak mózg* [Turing 1951*a*, s. 2]; wyróżnienie P.Ł.

Turing zwraca uwagę na fakt, że tym, co jest imitowane, są zachowania (funkcje) mózgu, nie zaś jego wewnętrzna struktura. W wywiadzie dla BBC z 1952 roku wyraził swoją opinię na ten temat w następujący, barwny sposób:

> Najważniejszą sprawą jest aby spróbować wytyczyć linię oddzielającą właściwości mózgu człowieka, o których chcemy dyskutować od tych, które nas nie interesują. Weźmy skrajny przypadek, nie interesuje mnie to, że mózg ma konsystencję zimnej owsianki. Nie powiemy przecież: „Ta maszyna jest całkiem twarda, czyli nie jest mózgiem, a więc nie może myśleć” [Newman et al. 1952, s. 3–4].

Przy takim rozumieniu naśladowania umysłu ludzkiego przez maszynę w naturalny sposób musimy się zgodzić, że nie będziemy oceniali jej wewnętrznej struktury, ale obserwowalne zachowanie maszyny, czy też, mówiąc dokładniej, stopień jego podobieństwa do zachowania człowieka w tej samej sytuacji. Turing pisze na ten temat:

> Jestem przekonany, że można skonstruować takie maszyny, które będą bardzo dokładnie symulowały działanie ludzkiego umysłu. Od czasu do czasu będą popełniały błędy i od czasu do czasu będą w stanie produkować nowe i interesujące stwierdzenia, zaś całość ich werbalnych zachowań będzie równie warta uwagi jak werbalne zachowania produkowane przez ludzki umysł [Turing 1951*b*, s. 2].

Jeżeli zgodzimy się z takim poglądem dotyczącym maszyn, jedyne czego potrzebujemy, to rodzaj testu (ze szczegółowo określonymi warunkami jego przeprowadzenia), który umożliwi nam badanie stopnia interesującego nas podobieństwa pomiędzy zachowaniem maszyny i człowieka. Zarys takiej propozycji znajdujemy już w raporcie Turinga napisanym w 1948 roku dla The National Physical Laboratory. Nosi on tytuł „Intelligent Machinery" i uważany jest za pierwszy manifest idei sztucznej inteligencji (por. [Copeland 2000], [Copeland, Proudfoot 2009]).[1] Tekst kończy opis pewnego eksperymentu, który w nieco zmodyfikowanej i uogólnionej formie przedstawi Turing w „Computing Machinery and Intelligence".

> Możliwe jest przeprowadzenie pewnego eksperymentu [...] nawet przy obecnym stanie wiedzy. Opracowanie papierowej maszyny[2], która całkiem dobrze radziłaby sobie z grą w szachy nie powinno być trudne. Do eksperymentu potrzebujemy trzech ludzi A, B, C. A i C powinni być raczej marnymi graczami szachowymi, zaś B będzie papierową maszyną (w celu zapewnienia rozsądnego tempa pracy byłoby dobrze, żeby był on zarówno matematykiem, jak i szachistą). Wykorzystujemy dwa pokoje, pomiędzy którymi zapewniono możliwość komunikacji. C gra albo z A, albo z papierową maszyną. W tej sytuacji może się okazać, że C będzie miał duże trudności z odgadnięciem z kim w rzeczywistości gra w szachy. (Jest to raczej wyidealizowana forma eksperymentu, który rzeczywiście przeprowadziłem) [Turing 1948, s. 37].

1.2. Rekonstrukcja oryginalnych założeń testu Turinga

Większość dyskusji i sporów wokół testu Turinga (TT) opiera się na najbardziej znanym z tekstów A.M. Turinga — „Computing Machinery and Intelligence" opublikowanym w czasopiśmie *Mind* w roku 1950. To właśnie w tym artykule Turing po raz pierwszy przedstawił w rozwiniętej formie ideę TT. W niniejszej analizie testu Turinga będę brał pod uwagę również następujące (mniej znane) teksty: „Intelligent Machinery" (raport dla The National Physical Laboratory z 1948), „Can Digital Computers Think" (wywiad radiowy wyemitowany w 1951 roku), „Intelligent Machinery, a Heretical Theory" (wykład wygłoszony w roku 1951), „Can automatic calculating machines be said to think?" (wywiad radiowy wyemitowany w 1952 roku) oraz „Digital Computers Applied to Games" (tekst z tomu *Faster than thought* pod redakcją B.V. Bowdena, który ukazał się w 1953 roku).

Test Turinga powstał na bazie pewnej gry towarzyskiej, nazywanej grą w naśladownictwo (*imitation game*). W grze tej biorą udział trzy osoby: mężczyzna (A), kobieta (B) i pytający — C (dowolnej płci). Mężczyzna i kobieta przebywają w osobnych pokojach, oddzieleni od siebie i od pytającego. Pytający może zadawać pytania osobom w pokojach, które to osoby zna jako

---

[1] Turing poruszył w nim zagadnienia, które dzisiaj zaliczane są do obszaru automatycznego dowodzenia twierdzeń, *problem solving*, algorytmów genetycznych oraz sztucznych sieci neuronowych (por. [Copeland 2000, s. 520], [Copeland, Proudfoot 2009, s. 120]).

[2] Pod pojęciem „papierowej maszyny" (*paper machine*) Turing rozumiał osobę wykonującą zadany algorytm przy użyciu kartki i ołówka (musimy pamiętać, że w 1948 roku nie było jeszcze komputerów, które mogłyby wykonywać np. algorytm gry w szachy).

X i Y. Jego zadaniem jest określenie (wyłącznie na podstawie uzyskanych odpowiedzi), w którym pokoju znajduje się kobieta, a w którym mężczyzna. Gracz A w trakcie gry ma robić wszystko, aby wprowadzić pytającego w błąd, tak aby ten dokonał nietrafnej identyfikacji — może zatem swobodnie kłamać, zaś B ma udzielać zawsze prawdziwych odpowiedzi.

Punktem wyjścia dla rozważań Turinga jest pytanie o to, czy maszyny mogą myśleć. Turing stwierdza jednak, że jest ono „[...] zbyt wieloznaczne, aby w ogóle zasługiwało na dyskusję" [Turing 1950, s. 422]. Dlatego właśnie pragnie on zastąpić to pytanie innymi — mianowicie, czy maszyna może odnieść sukces w zaprojektowanej przez niego grze.

> Zadajmy teraz pytanie „co stanie się, kiedy maszyna weźmie udział w tej grze"? Czy pytający dokona nietrafnej identyfikacji równie często, jak w przypadku, gdy w grze biorą udział kobieta i mężczyzna? [Turing 1950, s. 434].

Zdaniem Turinga, tak zaprojektowany test ma wyrównać szanse człowieka i maszyny, rozdzielając cielesność od zdolności intelektualnych. To, czego powinniśmy szukać, to analogie funkcjonalne między maszyną a człowiekiem.

Już w „Computing Machinery..." Turing wspomina o wersji TT, w którym bierze udział tylko dwóch uczestników i określa ją mianem *viva voce*. W okresie po ukazaniu się „Computing Machinery..." Turing mówi o TT tylko w kontekście tego typu gry. Wydaje się, że oddaje to podstawową intuicję związaną z TT — wystarczy, że sędzia (pytający) oceniał będzie jedynie odpowiedzi udzielane przez (potencjalny) komputer. Możemy więc pominąć trzeciego uczestnika gry bez szkody dla całej konstrukcji testu. W swoim artykule dotyczącym testu Turinga A. P. Saygin, I. Cicekli i V. Akman poświęcają sporo miejsca na dyskusję dotyczącą roli trzeciej osoby w TT (por. [Saygin et al. 2001]). Tymczasem sam Turing — w późniejszym okresie — wydaje się zupełnie ją ignorować, kładąc nacisk na inny aspekt gry w naśladownictwo. Robert Harnish w *Minds, Brains, Computers. An Historical Introduction to the Foundations of Cognitive Science* rozróżnia nawet grę, w której bierze udział trzech graczy od gry w wersji *viva voce*. Pierwszą z nich określa mianem „gry w naśladownictwo", zaś tylko tę drugą nazywa testem Turinga (por. [Harnish 2002, s. 183]).

W „Computing Machinery..." A. M. Turing w następujący sposób pisze o teście, w którym bierze udział dwóch graczy:

> Gra [w której pominięty został gracz B] jest często stosowana w praktyce pod nazwą *viva voce* aby odkryć, czy ktoś coś naprawdę zrozumiał, czy też tylko „wykuł to na blachę" [Turing 1950, s. 446].

Zaś w „Can Digital Computer Think":

> Myślę, że prawdopodobne jest na przykład to, że z końcem tego stulecia będziemy potrafili programować maszyny, aby odpowiadały na pytania w taki sposób, że będzie niesłychanie trudno zgadnąć, czy odpowiedzi udzielane są przez człowieka, czy przez maszynę. Wyobrażam sobie coś na kształt sprawdzania typu *viva voce*, ale z pytaniami i odpowiedziami, które są przesyłane w formie maszynopisu [...] [Turing 1951*a*, s. 4–5].

Przyjęcie tej wersji testu pozwala uprościć analizę bez utraty głównych intuicji związanych z TT sformułowanym dla trzech uczestników. Dlatego założenia testu wyszczególniane poniżej będą dotyczyły wersji *viva voce*:

1. W grze uczestniczy dwóch graczy: C (pytający, sędzia) oraz A (poddawany testowi).
2. Gracze nie mogą się widzieć, słyszeć, pisać do siebie pismem odręcznym.
3. To gracz C zadaje pytania, zaś gracz A na nie odpowiada.
4. Celem gracza poddawanego testowi jest wprowadzenie w błąd gracza sędziego tak, aby uniemożliwić mu trafną identyfikację (por. [Turing 1950, s. 434]). Gracz A zobowiązany jest również do postępowania zgodnie ze strategią mówiącą, że ma jak najlepiej naśladować odpowiedzi, jakich udzieliłby człowiek na jego miejscu. Zdaniem Turinga jest to najlepsza z możliwych do przyjęcia strategii, o czym świadczy poniższy fragment zaczerpnięty z „Computing Machinery...":

> Niektórzy mogą argumentować, że najlepszą strategią dla maszyny podczas „gry w naśladownictwo" mogłoby być coś innego niż naśladowanie zachowania człowieka. Być może tak jest, ale uważam, że jest mało prawdopodobne aby tego typu działania przyniosły jakiś znaczący efekt [...] Zakładam, że *najlepszą strategią w tej sytuacji jest udzielanie odpowiedzi takich, jakie w naturalny sposób udzielone zostałyby przez człowieka* [Turing 1950, s. 435]; wyróżnienie P.Ł.

Gracz A zgodnie z tą strategią może używać pewnych „trików": opóźniać nieco swoją odpowiedź (kiedy pytanie dotyczy np. zadania arytmetycznego), popełniać błędy w pisowni i błędy ortograficzne etc. O takiej możliwości wspomina Turing w „Can automatic calculating machines be said to think?":

> [...] maszyna będzie mogła stosować wszystkie rodzaje sztuczek, tak aby jawić się w bardziej ludzki sposób, takie jak czekanie zanim poda odpowiedź, czy popełnianie błędów ortograficznych [...] [Newman et al. 1952, s. 5].

A także w „Intelligent Machinery, a Heretical Theory":

> Jestem przekonany, że można skonstruować takie maszyny, które będą bardzo dokładnie symulowały działanie ludzkiego umysłu. Od czasu do czasu będą popełniały błędy i od czasu do czasu będą w stanie produkować nowe i interesujące stwierdzenia [...] [Turing 1951*b*, s. 2].

Odnośnie do pytań, które mogą pojawić się w ramach TT, Turing zdaje się nie wprowadzać szczególnych ograniczeń. Świadczą o tym poniższe fragmenty zaczerpnięte z „Computing Machinery..." oraz „Can automatic calculating machines be said to think?":

> Metoda pytań i odpowiedzi wydaje się być odpowiednia dla wprowadzenia niemal każdej z dziedzin ludzkiej aktywności, jaką chcielibyśmy rozważać [Turing 1950, s. 435].

> [Braithwaite:] Czy pytania musiałyby być działaniami do wykonania, czy też mógłbym zapytać co komputer zjadł na śniadanie?
> [Turing:] Tak, cokolwiek. [...] „Przecież ty tylko udajesz, że jesteś człowiekiem" byłoby zupełnie na miejscu [Newman et al. 1952, s. 5].

W „Computing Machinery..." Turing podaje dwa przykłady możliwego dialogu w sytuacji testu Turinga. Pierwszy z nich ma być ilustracją tego, że metoda pytań i odpowiedzi doskonale nadaje się do poruszenia niemal każdej dziedziny aktywności człowieka [Turing 1950, s. 434-435]:

C: Napisz mi proszę sonet na temat Forth Bridge.
A: Nie licz na mnie w tej sprawie. Nie jestem zbyt dobry w pisaniu poezji.
C: Dodaj 32957 i 70764
A: (Pauza około 30 sekund a później odpowiada) 105621
C: Czy grasz w szachy?
A: Tak.
C: Mam króla na e8 i nie mam innych figur. Ty masz tylko króla na e6 i wieżę na h1. Twój ruch. Jak zagrasz?
A: (Po piętnastominutowej przerwie) wieża na h8. Mat.

Drugi z przykładów dialogu to „wzorcowa" gra *viva voce* [Turing 1950, s. 446]:

C: Czy w pierwszym wierszu twojego sonetu, który brzmi „Czyż powinienem porównać cię do letniego dnia" sformułowanie „wiosenny dzień" nie byłoby równie dobre, czy może nawet bardziej odpowiednie?
A: Nie, to nie pasuje.
C: A co myślisz o „zimowym dniu". To pasowałoby tutaj całkiem dobrze.
A: Tak, ale nikt nie chce, żeby porównywać go z zimowym dniem.
C: Czy przyznałbyś, że pan Pickwick przywodzi ci na myśl święta Bożego Narodzenia?
A: Tak, przynajmniej w pewnym sensie.
C: Boże Narodzenie to zimowy dzień i nie sądzę, żeby pan Pickiwck miał coś przeciwko porównaniu, o którym rozmawiamy.
A: Myślę, że w tym momencie nie mówisz poważnie. Mówiąc o zimowym dniu mamy raczej na myśli typowy zimowy dzień, nie zaś tak wyjątkowy zimowy dzień, jakim są święta Bożego Narodzenia.

Na podstawie przytoczonych cytatów przyjmuję tutaj, że sędzia może formułować trzy rodzaje wypowiedzi: (*i*) pytania; (*ii*) zdania eksplikujące problem (powiązane z późniejszym pytaniem); (*iii*) zdania wyrażające opinie na temat udzielanych przez *A* odpowiedzi (przebiegu testu).

Szczególnie istotna z perspektywy tej pracy jest pewna propozycja dotycząca pytań, którą Turing wysunął w „Computing Machinery...". Jego zdaniem powinniśmy również rozważyć sytuację TT, w którym sędzia zadaje jedynie pytania rozstrzygnięcia. Turing pisze:

> Oczywiście zakładamy, na chwilę obecną, że mamy do czynienia raczej z pytaniami, na które adekwatną odpowiedzią jest „Tak" lub „Nie", niż pytaniami typu „Co sądzisz o Picassie?" [Turing 1950, s. 445].

Postulat taki miał zapewne źródło w niedoskonałości ówczesnych maszyn, ale można z niego wnioskować, że test mimo tego ograniczenia nadal spełniałby swoje funkcje. Wydaje się, że Turing — zdając sobie sprawę z wymagań stawianych przez TT — myślał o pewnego rodzaju ewolucji: najpierw maszyny byłyby w stanie podchodzić do TT, w którym miałyby do czynienia jedynie z pytaniami rozstrzygnięcia, dopiero w dalszej perspektywie pytania stawiane przez sędziego mogłyby być pytaniami bardziej wyrafinowanymi. Za współczesną realizację tej propozycji moglibyśmy uznać MIST (*Minimum Intelligent Signal Test*) autorstwa Chrisa McKinstry (por. [McKinstry 1997], [McKinstry 2009] oraz rozdział 4.1.3).

Warto również zaznaczyć, że TT pomyślany jest jako test o charakterze statystycznym i — zdaniem Turinga — powinien być on powtarzany kilkakrotnie dla uzyskania bardziej wiarygodnych wyników. Świadczą o tym przytoczone poniżej fragmenty.

> Główną ideą testu jest to, że maszyna — odpowiadając na zadawane jej pytania — ma udawać, że jest człowiekiem i zda ów test tylko wtedy, kiedy owo udawanie będzie przekonujące [...] Lepiej będzie jeśli założymy, że każdy z sędziów będzie musiał oceniać kilkakrotnie oraz, że czasami sędziowie będą mieli do czynienia rzeczywiście z człowiekiem a nie z maszyną. Zapobiegnie to temu, żeby za każdym razem stwierdzali oni bez zastanowienia „to jest maszyna" [Newman et al. 1952, s. 5].

> Jestem przekonany, że za około pięćdziesiąt lat możliwe stanie się takie programowanie komputerów z pojemnością pamięci rzędu 10^9, aby radziły sobie w grze w naśladownictwo tak dobrze, że przeciętny pytający nie będzie miał więcej niż 70 procent szans na dokonanie prawidłowej identyfikacji po pięciu minutach stawiania pytań [Turing 1950, s. 442].

Zagadnieniem o dużym znaczeniu dla konstrukcji TT jest również dobór sędziego. Ned Block w artykule „The Mind as the Software of the Brain" przekonuje, że pomysł testu jest chybiony właśnie z powodu niedoprecyzowania tego, w jaki sposób powinna być wybierana osoba pytająca. Zauważa on w szczególności, że „[...] ekspert [w dziedzinie komputerów — przyp. P.Ł.] może wiedzieć, że obecne inteligentne maszyny radzą sobie z pewnymi problemami kłopotliwymi dla ludzi" [Block 1995*b*, s. 379], dzięki czemu uzyska ogromną przewagę nad maszyną w teście Turinga. Turing zdawał sobie jednak sprawę z tej trudności. W „Computing Machinery..." pytający określany jest jako: „przeciętny pytający" (*average interrogator*) [Turing 1950, s. 442]; zaś w wywiadzie dla BBC jako osoba, która „[...] nie powinna być ekspertem w dziedzinie komputerów [...]" [Newman et al. 1952, s. 4]. Do dyskusji o roli doboru sędziego w teście Turinga powrócimy w rozdziale trzecim.

Podsumowując, na test Turinga możemy patrzeć jako na system pytań i odpowiedzi. Wydaje się, że przy konstruowaniu jego modelu możemy potraktować test Turinga jako pewną procedurę badawczą, przeprowadzaną przez gracza C, której celem jest ustalenie wiarygodności deklaracji gracza A, iż jest on człowiekiem. Aby przeprowadzić weryfikację tej deklaracji, sędzia sprawdza, jak gracz poddawany testowi „radzi sobie" w dziedzinach uznawanych za przejawy ludzkiej inteligencji. Test składa się więc z pewnych części — nazwijmy je rundami — z których każda dotyczy pewnego zakresu przejawów inteligencji. Dla każdej rundy sędzia formułuje problem, charakterystyczny dla określonego przejawu ludzkiej inteligencji.

Wyszczególnione powyżej założenia dotyczące testu Turinga (ich zestawienie zawiera tabela 1.1) rodzą kilka pytań i problemów. W szczególności intuicja dotycząca *problemu doboru sędziego* wyrażona przez przez N. Blocka wydaje się w tym kontekście bardzo interesująca. Wynik testu Turinga w dużej mierze będzie zależał od przekonań i poglądów sędziego — w skrajnej sytuacji może nawet zdarzyć się tak, że jeden z sędziów orzeknie o graczu A, że jest on maszyną, zaś inny sędzia (o tym samym graczu A), iż jest on człowiekiem.

Kolejnym ciekawym zagadnieniem jest *adekwatność odpowiedzi* udzielanych przez gracza poddawanego testowi. Po pierwsze, musimy zadecydować,

Tablica 1.1. Zestawienie oryginalnych założeń testu Turinga opisanych w rozdziale 1

Sytuacja testowa	– W grze bierze udział 2 graczy (sędzia i poddawany testowi). – Gracze nie mogą się widzieć ani słyszeć.
Przebieg testu	– Test ma charakter statystyczny. – Powinien być powtarzany kilkakrotnie. – Nie powinno być tak, że przy każdym powtórzeniu poddawany testowi jest maszyną. – TT kończy dokonana przez sędziego identyfikacja: „A jest maszyną”, „A jest człowiekiem”.
Sędzia	– Nie powinien być ekspertem w sprawie komputerów. – Może formułować trzy rodzaje wypowiedzi: pytania; zdania eksplikujące problem (powiązane z późniejszym pytaniem); zdania wyrażające opinie na temat udzielanych przez A odpowiedzi (przebiegu testu).
Pytania	– To sędzia zadaje pytania, a poddawany testowi na nie odpowiada. – W pierwszej kolejności lepiej jest rozważać TT z pytaniami rozstrzygnięcia. – Docelowo w TT powinny być dozwolone pytania bardziej wyrafinowane (dotyczące każdej ze sfer ludzkiej aktywności).
Odpowiedzi	– Poddawany testowi ma jak najlepiej naśladować odpowiedzi udzielane przez człowieka. – Może używać „trików”: opóźniać odpowiedź (np. gdy jest to zadanie arytmetyczne), popełniać błędy itp.

kiedy reakcja gracza A na zadane pytanie może zostać uznana za odpowiedź na owe pytanie. Ponadto pozostaje jeszcze problem zalecenia Turinga, że odpowiedzi udzielane przez maszynę mają być takie, „jakich udzieliłby człowiek” w rozważanej sytuacji — musimy ustalić kryterium, które pozwoli na zdecydowanie, czy dana odpowiedź jest taka, jakiej udzieliłby człowiek.

W związku z postulatem o *statystycznym charakterze testu* również powstaje wiele ciekawych pytań. Ile razy należy przeprowadzać test dla jednego gracza? Jaki powinien być stosunek trafnych i nietrafnych identyfikacji (ewentualnie ile powinno być trafnych), żeby można było powiedzieć, że gracz A zdał TT? Ile powinien trwać jeden test? To tylko niektóre z nich.

W dużej mierze rozważania przedstawione w dalszej części tej książki stanowią próbę zmierzenia się właśnie z wymienionymi tutaj problemami.

1.3. Co bada test Turinga?

Dla Turinga zastąpienie pytania „czy maszyny mogą myśleć?” pytaniem o powodzenie maszyn w pewnej zaprojektowanej przez niego grze nie miało być kluczem do rozstrzygnięcia problemu bycia istotą inteligentną w ogóle. W wywiadzie dla BBC z 1952 roku Turing, po krótkim omówieniu propozycji testu, powiedział:

> A więc tak przedstawia się mój test. Oczywiście, na chwilę obecną, nie twierdzę ani że maszyny rzeczywiście mogą go zdać, ani że nie. Moja sugestia dotyczy raczej tego, że jest to właśnie pytanie, o którym warto dyskutować. *Nie jest ono identyczne* z pytaniem „Czy maszyny myślą", ale — *zważywszy na nasz cel* — wydaje się ono wystarczająco do niego zbliżone [Newman et al. 1952, s. 5–6]; wyróżnienia P.Ł.

Test Turinga nie ma zatem spełniać roli *definicji* własności bycia istotą inteligentną. Co do tego zgadza się większość badaczy testu Turinga. W literaturze dotyczącej TT znajdujemy natomiast szeroko zakrojoną dyskusję dotyczącą tego, czy test Turinga możemy potraktować jako implikujący operacyjną (a więc cząstkową) definicję posiadania inteligencji. Argumenty na rzecz takiej interpretacji TT przedstawiają m.in. N. Block [Block 1995*a*, s. 248], R. French [French 2000, s. 115] [French 1990, s. 53], A. Hodges [Hodges 1992, s. 415], P. H. Millar [Millar 1973, s. 595] czy J. Searle [Searle 1980, s. 423]. Polemikę z takim stanowiskiem znajdziemy np. w [Copeland 2000], [Moor 1976].

Część autorów (m.in. N. Block i J. Searle), uznając, że TT ma służyć operacyjnej definicji posiadania inteligencji przez maszyny, na tej podstawie krytykuje go za skrajnie behawiorystyczny charakter takiej definicji. Wydaje się jednak, że warto pokusić się na spojrzenie na to zagadnienie z nieco innej perspektywy, która — naszym zdaniem — znacznie lepiej pasuje do intencji samego Turinga. Perspektywą tą jest (klasyczne już na gruncie psychologii) rozróżnienie na inteligencję A, B i C wprowadzone przez D. O. Hebba w 1949 roku (por. [Strelau 1987], [Nęcka 2005]). Pod pojęciem inteligencji A rozumie się wrodzone możliwości; zaś pod pojęciem inteligencji B, możliwości rzeczywiście rozwinięte. Natomiast inteligencja C „ogranicza się do zachowań, które ujawniają się w badaniach na podstawie testów inteligencji" [Strelau 1987, s. 17]. Tak rozumiane pojęcie inteligencji C ma charakter operacjonalistyczny. Nie sprawia to jednak, że jest ona postrzegana jako mniej wartościowa — stanowi integralną część badań nad inteligencją człowieka. E. Nęcka w następujący sposób opisuje wzajemne relacje pomiędzy inteligencją A, B i C:

> Tylko część wrodzonych możliwości (inteligencja A) rozwija się w postaci inteligencji B, pozostała część rzeczywistych uzdolnień (inteligencja B) wynika z wiedzy i doświadczenia, a nie w wrodzonych zadatków. Inteligencja B jest tyko w części wykrywana za pomocą testów, ujawniając się jako inteligencja C. Natomiast pewna część wariancji wyników testowych (inteligencji C) nie zależy ani od A, ani od B, a na przykład od lęku przed oceną lub nieumiejętności zdawania testów [Nęcka 2005, s. 22].

Wydaje się, że do takiego spojrzenia na zagadnienie badania inteligencji pasują intencje Turinga. We wspomnianym wywiadzie dla BBC na pytanie o to, czy dysponuje jakąś definicją bycia istotą inteligentną stwierdza on bowiem:

> *Nie chcę podawać definicji myślenia*, ale jeśli bym musiał, to prawdopodobnie nie byłbym w stanie powiedzieć nic ponad to, że jest to coś w rodzaju brzęczenia (*buzzing*), które zachodzi w mojej głowie. Nie sądzę jednak abyśmy musieli w ogóle zgadzać się co do jakiejkolwiek definicji. Najważniejszą sprawą jest aby spróbować wytyczyć linię oddzielającą właściwości mózgu człowieka, o których chcemy dyskutować od tych, które nas nie interesują. [...] Chciałbym zaproponować pewien test, który można by zastosować do maszyn. Można by nazwać go testem sprawdzającym, czy maszyna myśli.

> Ale lepiej byłoby uniknąć niepotrzebnych dyskusji i powiedzieć, że *maszyny, które pomyślnie przechodzą ów test są (powiedzmy) maszynami Klasy A* [Newman et al. 1952, s. 3–4]; wyróżnienia P.Ł.

W tym kontekście ciekawe wydaje się pytanie o to, jakiego rodzaju świadectwa/uzasadnienia dla tezy o posiadaniu przez maszyny inteligencji dostarcza sukces odniesiony przez maszynę w teście Turinga. Najbardziej interesujące będą dla nas propozycje Jamesa Moora i Douglasa Stalkera. Pierwszy z nich przekonuje, że uzasadnienie to ma charakter indukcyjny (por. [Moor 1976, 1978, 2001], por. też [Watt 1996]).

> Jestem przekonany, że tym, co jest istotne w teście Turinga, jest fakt, że dostarcza on dobrego wzorca dla gromadzenia indukcyjnych uzasadnień. Dzięki temu, gdyby test Turinga został zdany, mielibyśmy adekwatne podstawy dla indukcyjnego rozumowania prowadzącego do wniosku, że komputer może myśleć na poziomie normalnego dorosłego człowieka [Moor 1976, s. 299–300].

Punktem wyjścia dla Moora jest zadanie pytania o to, w jaki sposób my, jako ludzie, nabywamy przekonania o tym, że inni ludzie myślą. Jego zdaniem takie przekonanie jest częścią większej teorii, którą budujemy w celu wyjaśnienia zachowań innych. Oczywiście procesy myślowe są bardzo złożone i można je rozpatrywać w wielu aspektach, ale niezaprzeczalnym faktem jest to, że nie mamy bezpośredniego dostępu do cudzych stanów mentalnych. Jedyne do czego mamy dostęp, to szeroko rozumiane zachowania innych ludzi. Na ich podstawie indukcyjnie wyprowadzamy wspomnianą teorię. Obserwacja zachowań innych pozwala nam na potwierdzanie, negowanie i modyfikacje wyprowadzonej przez nas teorii. W takiej sytuacji „nie ma wyraźnego powodu, dla którego wiedza o myśleniu komputerów nie miałaby powstawać w taki sam sposób" [Moor 1976, s. 299].

Na pierwszy rzut oka propozycja Moora wydaje się bardzo intuicyjna i przekonująca. D. F. Stalker zauważa jednak pewną jej słabość:

> Zgodnie z tym [tzn. Moora — przyp. P.Ł.] podejściem nasze przekonania co do sfery mentalnej innych osób są częścią pewnej teorii wyjaśniającej. W celu wyjaśnienia zachowań innych ludzi powołujemy się na teorię, która wymaga użycia pojęcia myślenia. Ale to nie daje nam pełnego obrazu. Nie mówi nam *dlaczego* powinniśmy przyjmować, że zachowania innych osób stanowią świadectwo posiadania przez nie jakiegoś rodzaju życia umysłowego [Stalker 1976, s. 308]; wyróżnienie P.Ł.

Zdaniem Stalkera, przyjmujemy taką a nie inną teorię przypisywania innym stanów mentalnych, ponieważ jest ona najlepszą posiadaną przez nas teorią. Przy jej wyborze posługujemy się więc nie tyle indukcją, ile rozumowaniem przypominającym rozumowanie abdukcyjne.

Schemat rozumowania abdukcyjnego możemy przedstawić, za Peircem, w sposób następujący (por. [Peirce 1931/1958]):

Obserwujemy zaskakujące zjawisko P.
Gdyby Q było prawdziwe, zachodzenie P byłoby oczywistością.
———————————————
Mamy zatem podstawy, by podejrzewać, że Q jest prawdziwe.

Warto wspomnieć, że istnieją dwie interpretacje rozumowań abdukcyjnych (por. [Urbański 2005, s. 146]). Wedle jednej z nich abdukcja służy jedynie generowaniu zbiorów hipotez wyjaśniających. Wedle drugiej interpretacji służy

nie tylko generowaniu, ale również ocenie tych hipotez wyjaśniających. Stalker wydaje się właśnie zwolennikiem tej drugiej interpretacji.

Wyboru teorii wyjaśniającej zachowania maszyn dokonujemy podobnie, jak wyboru teorii pozwalającej nam wyjaśniać zachowania innych ludzi. Jeśli przyjmiemy, że w powyższych schematach P oznaczać będzie zdanie TT, zaś Q bycie inteligentnym, to z tego, że maszyna zdaje TT razem z poglądem, że inteligencja (a przynajmniej jakiś jej rodzaj) implikuje zdolność do zdania TT wnioskujemy, że poddawana testowi maszyna posiada inteligencję (por. [Shieber 2007]).

A zdaje test Turinga.
Jeżeli A jest inteligentny, to A zdaje test Turinga.
Mamy zatem podstawy, by podejrzewać, że A posiada inteligencję.

Uznanie, że sukces odniesiony w teście Turinga dostarcza jedynie abdukcyjnego uzasadnienia dla tezy o posiadaniu inteligencji przez maszyny może być postrzegane jako zbytnie osłabienie propozycji Turinga. Wydaje się jednak, że takie ujęcie problemu uzasadniania dostarczanego przez TT bardzo dobrze odpowiada — opisanej powyżej — procedurze przyjmowanej przez Turinga przy konstruowaniu jego propozycji, tj. zastąpieniu pytania „czy maszyna może myśleć?” pytaniem o powodzenie maszyny w przeformułowanej grze w naśladownictwo.

1.4. Pewne nieporozumienia wokół testu Turinga

W literaturze przedmiotu można znaleźć pewne dyskusje wokół TT, które wydają się mieć źródło w niezrozumieniu istoty propozycji Turinga. Omówię tutaj trzy z nich: dyskusję poświęconą przewidywaniom Turinga co do czasu powstania maszyn myślących; próby traktowania testu Turinga jako testu płci; a także zarzuty dotyczące antropocentryzmu testu Turinga.

1.4.1. Prognoza Turinga

Turing w „Computing Machinery...” zawarł, cytowaną już wyżej, następującą prognozę dotyczącą możliwości powstania maszyn zdolnych do zdania testu Turinga:

> Jestem przekonany, że za około pięćdziesiąt lat możliwe stanie się takie programowanie komputerów z pojemnością pamięci rzędu 10^9, aby radziły sobie w grze w naśladownictwo tak dobrze, że przeciętny pytający nie będzie miał więcej niż 70 procent szans na dokonanie prawidłowej identyfikacji po pięciu minutach pytania [Turing 1950, s. 442].

Wielu autorów, powołując się na powyższy fragment „Computing Machinery...”, krytykuje Turinga za zbytni optymizm, a nawet wyprowadza z tej wypowiedzi argument przeciwko testowi Turinga, traktując przedstawione w niej przewidywania jako coś, co do czego Turing był całkowicie przeko-

nany (por. [Purtill 1971], [Sampson 1973], [Whitby 1996], [Whitby 1997], [Saygin et al. 2001]). G. Sampson pisze po prostu:

> Turing zaproponował „grę w naśladownictwo” jako kryterium pozwalające zdecydować, czy komputer może myśleć i *przewidział, że do roku 2000* niektóre komputery pomyślnie przejdą zaproponowany test [Sampson 1973, s. 173]; wyróżnienie P.Ł.

Inny badacz zajmujący się testem Turinga — R. Purtill — zdecydowanie krytykuje Turinga za przewidywania dotyczące maszyn pomyśnie przechodzących test. W „Beating the Imitation Game” pisze:

> [...] przewidywanie Turinga dotyczące tego, że komputery, które mogłyby zagrać w grę w naśladownictwo zbudowane zostaną w przeciągu pięćdziesięciu lat od ukazania się jego artykułu (*to znaczy w roku 2000*) było w oczywisty sposób dużą przesadą [Purtill 1971, s. 169]; wyróżnienie P.Ł.

Tymczasem, jak słusznie zauważają zarówno S. Shieber [Shieber 2004, s. 98], jak i B. J. Copeland [Copeland 2000, s. 527] jest to tylko jedno z przewidywań Turinga odnośnie do tej sprawy i na pewno nie należy traktować tej jego wypowiedzi jako swego rodzaju prognozy, która — niespełniona — może służyć za podstawę krytyki testu Turinga.

W przytoczonym wcześniej fragmencie „Can digital computers think?” Turing podkreśla, że wszelkie tego typu prognozy są jedynie jego przypuszczeniami:

> Myślę, że prawdopodobne jest na przykład to, że z końcem tego stulecia będziemy potrafili programować maszyny, aby odpowiadały na pytania w taki sposób, że będzie niesłychanie trudno zgadnąć czy odpowiedzi udzielane są przez człowieka, czy przez maszynę. [...] *Przedstawiam tu jedynie moje zdanie w tej sprawie; jest jeszcze wiele do powiedzenia dla innych* [Turing 1951*a*, s. 4–5]; wyróżnienie P.Ł.

W wywiadzie dla BBC z 1952 roku znajdujemy już znacznie bardziej ostrożną prognozę:

> [Newman:] Chciałbym być przy tym, kiedy zostanie rozegrany ów mecz między człowiekiem a maszyną i spróbować swoich sił w formułowaniu niektórych pytań. Zdaje się jednak, że jeżeli żadne z pytań nie będą zabronione, minie sporo czasu zanim maszyny będą miały choć cień szansy.
> [Turing:] O tak, powiedziałbym, że przynajmniej 100 lat [Newman et al. 1952, s. 6].

Należy pamiętać, że podane fragmenty wypowiedzi Turinga stanowią jedynie jego przypuszczenia (co wydaje się zupełnie zrozumiałe, jeśli weźmiemy pod uwagę to, że w momencie ich formułowania nie istniały jeszcze komputery w dzisiejszym tego słowa znaczeniu). Wykorzystywanie tych prognoz do krytykowania idei testu Turinga wydaje się sporym nadużyciem. Istnieje wiele słabych punktów propozycji Turinga — zwłaszcza jeżeli chcemy ją odtworzyć jedynie na podstawie „Computing Machinery...” — i to nimi należałoby się raczej zająć. Dyskusja o tym czy Turing był, czy nie był zbytnim optymistą, czy jego „przepowiednia” sprawdziła się, czy nie, nie wnosi wartościowych wątków do dyskusji dotyczących adekwatności testu Turinga jako narzędzia rozpoznawania obecności inteligencji w sztucznych systemach poznawczych.

1.4.2. Test Turinga jako test płci

Podobny charakter ma dyskusja dotycząca nieprawidłowego, zdaniem niektórych autorów, odczytania propozycji Turinga z „Computing Machinery. . . ". W celu usystematyzowania stanowisk w tej sprawie powstał nawet podział na *standardowy* sposób odczytywania tekstu Turinga oraz na sposób *literalny*. Sformułowanie tego podziału znajdujemy w artykule „Turing's rules for the Imitation Game" autorstwa G. Piccininiego:

> Przy standardowym odczytaniu, test Turinga można skrótowo opisać jako porównanie ludzi i maszyn, w którym pytający wymaga od maszyny zademonstrowania odpowiedniej biegłości w posługiwaniu się ludzkim językiem, wiedzą oraz zdolnościami rozumowania. Opanowanie tych umiejętności [...] stanowi wyraźny znak inteligencji czy też myślenia [Piccinini 2000, s. 572].

Jeżeli zaś chodzi o zwolenników literalnego odczytywania „Computing Machinery..." (por. [Genova 1994], [Lassègue 1996], [Lassègue 2009], [Naur 1986], [Gelernter 1994], [Hayes, Ford 1995]), to:

> [...] sugerują, że celem maszyny jest symulowanie *mężczyzny imitującego kobietę*, podczas gdy pytający — nieświadomy rzeczywistego celu testu — nadal stara się określić, który z dwojga graczy jest kobietą, a który mężczyzną [Piccinini 2000, s. 572].

Uzasadnienia szukają oni we fragmencie „Computing Machinery...", w którym Turing najpierw opisuje grę w naśladownictwo, po czym zastępuje jednego z graczy (mężczyznę) maszyną. Zadaniem komputera nie jest więc naśladowanie człowieka jako takiego, ale raczej człowieka konkretnej płci. Jak pisze Judith Genova:

> [...] test zdolności maszyny do myślenia okazuje się nie dotyczyć tego, czy maszyna jest w stanie przekonać sędziego, który jest człowiekiem, że ona również nim jest, ale raczej tego, żeby zwieść gracza C, tak aby przekonany był, że maszyna jest człowiekiem określonego rodzaju, tzn. raczej mężczyzną, niż kobietą [Genova 1994, s. 313–314].

Argumentacja zwolenników literalnego odczytywania propozycji Turinga opiera się na założeniu, że takie, a nie inne sformułowanie odnośnego fragmentu „Computing Machinery..." nie było w tym miejscu przypadkowe. Na uwagę zasługują tutaj teksty J. Genovy oraz J. Lassègue'a. Autorzy ci, opierając się na biografii Turinga starają się doszukać jej wpływu na treść „Computing Machinery...". Dochodzą do wniosku, że dla Turinga inteligencja była w jakiś sposób zależna od płci (w obu tekstach znajdziemy np. twierdzenie, że Turing uważał kobiety za mniej inteligentne od mężczyzn). Z kolei zdaniem Susan Sterrett, naśladowanie płci przez maszynę w grze w naśladownictwo ma sprawić, że TT będzie trudniejszy i bardziej wiarygodny jako narzędzie badania maszynowej inteligencji (por. [Sterrett 2000]).

Zarówno A. Hodges (por. [Hodges 1998]), jak i S. Shieber (por. [Shieber 2004]) zwracają uwagę na to, że rzeczywiście modyfikacja wprowadzona do gry w imitację przez Turinga może być na pierwszy rzut oka myląca, ale dalsza lektura „Computing Machinery..." — zwłaszcza w momencie, gdy Turing wprowadza grę typu *viva voce* — powinna usunąć wszelkie wątpliwości. Literalne odczytanie tekstu „Computing Machinery..." jest również nie

do utrzymania w konfrontacji z przeprowadzoną powyżej rekonstrukcją TT w oparciu o inne teksty Turinga dotyczące proponowanego przez niego testu (por. rozdział 1.2).

1.4.3. Antropomorfizm testu Turinga

Zarzut o antropomorficznym charakterze testu Turinga pojawia się stosunkowo często w literaturze przedmiotu (por. np. [Saygin et al. 2001, s. 467–468], [Copeland, Proudfoot 2009, s. 128–129], [Drozdek 1998], [Cullen 2009]). Jedno z lepiej znanych jego sformułowań znajdujemy w artykule R. Frencha „Subcogniton and the Limits of the Turing Test" [French 1990]. French podkreśla w nim, że test Turinga jest testem wyłącznie (i typowo) ludzkiej inteligencji, co czyni go nieciekawą propozycją narzędzia do rozpoznawania obecności inteligencji w sztucznych systemach poznawczych.

Wydaje się, że krytyka testu Turinga, jako testu zorientowanego antropocentrycznie, wynika z niezrozumienia idei propozycji Turinga. Jak ujmują to B. J. Copeland i D. Proudfoot:

> Zamierzeniem Turinga było dokładnie to, aby gra w naśladownictwo testowała, czy dana maszyna emuluje — lub nie — inteligentne zachowania *ludzkiego* mózgu [Copeland, Proudfoot 2009, s. 129]; wyróżnienie P.Ł.

Na korzyść takiego stanowiska może przemawiać to, jaką strategię postępowania w teście zaleca Turing dla testowanej maszyny. W „Computing Machinery..." podkreśla on, że:

> Niektórzy mogą argumentować, że najlepszą strategią dla maszyny podczas „gry w naśladownictwo" mogłoby być coś innego niż naśladowanie zachowania człowieka. Być może tak jest, *ale uważam, że jest mało prawdopodobne aby tego typu działania przyniosły jakiś znaczący efekt* [...] [Turing 1950, s. 435]; wyróżnienie P.Ł.

Dlatego właśnie postuluje on, aby maszyna biorąca udział w teście imitowała ludzkie zachowania, udzielając odpowiedzi takich „[...] jakie w naturalny sposób udzielone zostałyby przez człowieka" [Turing 1950, s. 435]. Cała idea testu Turinga opiera się na prostym w zasadzie pomyśle — zastąpmy pytanie „czy maszyny mogą myśleć?" takim pytaniem, co do którego istnieje choć cień szansy na udzielenie jednoznacznej odpowiedzi (a więc pytaniem o to, czy maszyna poradzi sobie w pewnego typu grze *równie dobrze jak człowiek*). Formułowanie zarzutu o antropomorficznym charakterze testu Turinga wydaje się w tym kontekście nietrafne. Próby „wzbogacenia" TT o rozmaite dodatkowe aspekty, które mógłby on badać stoją w sprzeczności z ideą zaprojektowania w miarę prostego narzędzia, które daje nadzieję na praktyczne wykorzystanie (por. choćby próby „utrudnienia" testu Turinga opisane w rozdziale drugim w kontraście z systemami CAPTCHA przedstawionymi w rozdziale czwartym niniejszej książki).

Ciekawy głos w tej dyskusji zabiera również A. Drozdek w artykule „Human Intelligence and Turing Test". Pisze on:

> Komputery zostały skonstruowane i zaprogramowane przez ludzi; a zatem to ludzka inteligencja została wykorzystana aby je stworzyć. Tym samym [...] możemy komunikować się z komputerami, ponieważ są one owocem ludzkiej inteligencji i ludzka inteligencja jest w nie niejako wbudowana. Komunikacja jest możliwa, ponieważ stworzyliśmy je do komunikowania się, ponieważ komputery zaprogramowane są właśnie do komunikowania się — a przynajmniej tak, aby sprawiały wrażenie, że się komunikują [Drozdek 1998, s. 317].

Stanowisko takie wspierają wyniki badań, prowadzonych w celu bliższego zrozumienia interakcji ludzi z komputerami. W wynikach tych często uwidacznia się antropomorfizujący stosunek do komputerów (niektórzy badacze uważają nawet, że jest to domyślne nastawienie poznawcze ludzi w odniesieniu do tych elementów środowiska, których nie jesteśmy w stanie w pełni kontrolować — por. [Caporael, Heyes 1996]). Okazuje się, że ludzie w trakcie interakcji z komputerem — czy może lepiej powiedzieć z programem komputerowym — wykazującym choćby elementarne umiejętności konwersacyjne, stosują reguły społeczne znane z interakcji z innymi ludźmi (por. m.in. [De Angeli et al. 1999], [Dryer 1999], [De Angeli, Lynch, Johnson 2001], [De Angeli, Graham, Johnson, Coventry 2001], [Gratch, Marsella 2005] oraz [van Vugt et al. 2007]).

Rozdział 2

Spory i dyskusje wokół testu Turinga

2.1. Wprowadzenie

U podłoża propozycji A.M. Turinga leżą bardzo silne intuicje dotyczące tego, w jaki sposób nabieramy przekonania o obecności stanów mentalnych u innych osób (por. rozdział 1.3). Atrakcyjna wydaje się również względna prostota dostarczanego przez test Turinga kryterium posiadania inteligencji przez maszyny (uwidacznia się ona szczególnie w kontekście propozycji uczynienia TT bardziej restrykcyjnym, omówionych w dalszej części tego rozdziału). Zgodność z pewnymi intuicjami nie może być jednak ostatecznym wyznacznikiem adekwatności propozycji rozwiązania problemu tak ważkiego, jak ten wyrażony pytaniem: czy maszyny mogą myśleć? Dlatego też wokół zagadnienia adekwatności testu Turinga, jako narzędzia służącego do rozpoznawania obecności inteligencji w sztucznych systemach poznawczych (jak będę starał się pokazać poniżej, rozumianego przynajmniej dwojako), toczą się żywe dyskusje od momentu jego powstania aż po dzień dzisiejszy.

Pierwszą grupę zarzutów wobec testu zamieszcza Turing już w „Computing Machinery..." Są to: (1) sprzeciw teologiczny, (2) argument „głów schowanych w piasku", (3) argument matematyczny, (4) argument ze świadomości, (5) argument z różnych niemożności, (6) zarzut lady Lovelace, (7) argument z ciągłości systemu nerwowego, (8) argument z nieformalności zachowania oraz (9) argument z percepcji pozazmysłowej. Większość z nich ma dziś znaczenie jedynie historyczne, niemniej jednak zasługują one na choć krótkie omówienie. Niektóre z przedstawionych argumentów wydają się z dzisiejszej perspektywy nieaktualne bądź nawet dziwne (tak jak argument z percepcji pozazmysłowej). Część z nich jest jednak nadal obecna w literaturze przedmiotu lub powraca w niej w nieco zmienionej postaci (tak jest na przykład z zarzutem lady Lovelace oraz z zarzutem matematycznym).

1. Obiekcję natury teologicznej (*The Theological Objection*) rekonstruuje Turing w następujący sposób: „Myślenie jest funkcją ludzkiej nieśmiertelnej duszy. Bóg ofiarował duszę każdemu mężczyźnie i każdej kobiecie, ale nie obdarzył nią zwierząt ani maszyn. Dlatego też ani zwierzę, ani maszyna nie mogą myśleć" [Turing 1950, s. 443].

Turing nie zgadza się z żadną częścią tego rozumowania, ale podejmuje próbę odpowiedzi w podobnym stylu: tak skonstruowany argument godzi we wszechmoc Bożą. Moglibyśmy sobie przecież wyobrazić, że Bóg decyduje się obdarować duszą słonia, a nawet ... maszynę.

2. Argument „głów schowanych w piasku” (*The ‘Heads in the Sand’ Objection*): „Konsekwencje myślenia maszyn będą zbyt nieprzewidywalne, miejmy więc nadzieję, że nie będą one tego robiły” [Turing 1950, s. 444]. Argument ten, zdaniem Turinga, nie wymaga odrzucenia czy obalenia, ponieważ ma swoje podłoże w reakcji emocjonalnej. Spełnia on jednak ważną rolę, wskazując na pewne niebezpieczeństwa, jakie niesie ze sobą idea myślących maszyn. Nie powinien jednak prowadzić do skrajnych postaw, bowiem nieznajomość konsekwencji wynikających ze skonstruowania myślącej maszyny wcale nie pociąga za sobą odgórnej konieczności zrezygnowania z projektu skonstruowania sztucznej inteligencji. Argument „głów schowanych w piasku” pojawia się w literaturze przedmiotu w dyskusjach poświęconych tzw. sztucznemu geniuszowi (por. np. [Penrose 2000], [Penrose 1995], [Penrose 2001], [Lem 1999], [Marciszewski 1995], [Łupkowski 2005*b*]) oraz w szerzej zakrojonych rozważaniach dotyczących oceny postępu technologicznego (por. np. „Etyka technologii i technologia etyki” w [Lem 1984]; por. też [Jonas 1996], [Kiepas 1992], [Putnam 1975], [Horgan 1999]).

3. Argument matematyczny (*The Mathematical Objection*). Istnieją wyniki z dziedziny logiki matematycznej, które wskazują na pewne ograniczenia maszyn cyfrowych. Ze względu na te ograniczenia, maszyna biorąca udział w teście Turinga nie będzie w stanie odpowiedzieć na pewne pytania lub udzieli odpowiedzi, która jest błędna (a tym samym zdemaskuje siebie jako maszynę).
   Zdaniem Turinga, argument ten będzie groźny dla testu Turinga dopiero, gdy uda się wykazać, że podobne ograniczenia nie dotyczą ludzkiego intelektu. Do argumentu matematycznego wrócimy w dalszej części niniejszego rozdziału, przedstawiając jego wersję sformułowaną za pomocą logiki pytań.

4. Argument ze świadomości (*The Argument from Consciousness*). Zarzut ten można rozbić na dwie części. W pierwszej z nich mówi się, że maszyny cyfrowe nie będą mogły myśleć, ponieważ aby myśleć, trzeba *wiedzieć*, że się myśli, czyli posiadać pewną formę samoświadomości. W drugiej części twierdzi się, że myślenie nieodłącznie związane jest z okazywaniem całej gamy emocji, do czego nie byłyby zdolne maszyny cyfrowe.
   Turing zauważa, że pierwsza część argumentu może prowadzić do zajęcia stanowiska solipsystycznego. Odnośnie do drugiej części argumentu ze świadomości, Turing przytacza hipotetyczną rozmowę maszyny z człowiekiem, w której maszyna korzysta z programu na tyle zaawansowanego, że przechodzi test Turinga. Na podstawie tej rozmowy Turing stara się pokazać, że w odpowiedziach udzielanych przez dysponującą wystarczająco wyrafinowanym programem maszynę można odnaleźć przejawy rozmaitych emocji.

5. Argument z różnych niemożności (*Arguments from Various Disabilities*) opiera się na rozumowaniu następującym: „Gwarantuję ci, że pomimo tego, że jesteś w stanie zbudować maszyny o wszystkich wymienionych przez ciebie zdolnościach, to nigdy nie zbudujesz maszyny, która może zrobić X" [Turing 1950, s. 447], gdzie za X można podstawić np.: bycie towarzyskim, posiadanie poczucia humoru, posiadanie ulubionych potraw itp. Zdaniem Turinga taka postawa może wynikać z niepełnej wiedzy o zasadach działania maszyn cyfrowych. Zgadza się on z tym, że współczesne mu maszyny rzeczywiście nie posiadają imponujących możliwości imitowania typowo ludzkich zachowań, ale zauważa jednocześnie, że wzrost możliwości obliczeniowych komputerów prawdopodobnie zlikwiduje ten problem.
6. Zarzut lady Lovelace (*Lady Lovelace's Objection*): maszyna cyfrowa jest w stanie zrobić tylko to, co nakazuje jej program, nie ma tu miejsca na inwencję twórczą. Turing przyznaje, że maszyny są ograniczone programami, zgodnie z którymi działają, ale równie dobrze można powiedzieć, że człowiek posiada podobne ograniczenia (wynikające z jego budowy, genów, wiedzy itp.). Jeżeli wyobrazimy sobie wystarczająco skomplikowany program (uruchomiony na maszynie o odpowiednio dużych możliwościach), to ograniczenia, jakim będzie on podlegał mogą być bardziej podobne do tych, którym podlega człowiek niż do tych, którym podlegają proste programy.
   Zarzut lady Lovelace posłużył za inspirację do stworzenia tzw. testu lady Lovelace (*Lovelace Test*), który zdaniem jego autorów lepiej nadaje się do rozpoznawania obecności inteligencji w sztucznych systemach poznawczych niż test Turinga. Test lady Lovelace opisany jest szczegółowo w rozdziale 4.1.2.
7. Argument z ciągłości układu nerwowego (*Argument from Continuity in the Nervous System*): układ nerwowy nie może być modelowany przez maszynę o stanach dyskretnych, ponieważ ma charakter analogowy. Turing odpowiada jedynie, że w sytuacji testu Turinga różnica pomiędzy maszyną o stanach dyskretnych i maszyną analogową jest bez znaczenia — sędzia nie będzie w stanie w żaden sposób wykorzystać wiedzy o tej różnicy w celu dokonania trafnej identyfikacji gracza.
8. Argument z nieformalności zachowania (*The Argument from Informality of Behaviour*). Argument ten opiera się na założeniu, że nie jesteśmy w stanie spisać wszystkich możliwych reguł zachowania dla wszystkich możliwych sytuacji, które mogłyby mieć miejsce. Turing zauważa, że argument ten jest niejasny. Należałoby wyjaśnić, czy wspomniane reguły mówią, jak zachowa się człowiek w pewnej sytuacji, czy też jak *powinien* się zachować. Jeżeli rozważamy drugą z opcji, to nie ma żadnych podstaw, aby twierdzić, że maszyny byłyby tutaj w gorszej sytuacji niż ludzie. Nie jest bowiem możliwe sformułowanie kompletnego kodeksu oczekiwanych zachowań dla każdej możliwej sytuacji — dotyczy to zarówno ludzi, jak i maszyn.
   Argument z nieformalności zachowania wpisuje się w szerszą dyskusję dotyczącą tzw. problemu ramy (*frame problem*) w kontekście badań nad

sztuczną inteligencją, a także rozważań epistemologicznych[1]. Obszerne omówienie tego zagadnienia w odniesieniu do testu Turinga zawiera monografia autorstwa L. J. Crocketta *The Turing Test and the Frame Problem. AI's Mistaken Understanding of Intelligence* [Crockett 1994].

9. Argument z percepcji pozazmysłowej (*Extra Sensory Perception*). Zjawiska zaliczane do dziedziny percepcji pozazmysłowej (takie jak: telepatia, jasnowidzenie, psychokineza) nigdy nie będą dostępne maszynie cyfrowej, ponieważ wszystkie zdają się przeczyć temu, co moglibyśmy umieścić w ramach wyjaśnienia naukowego. Gdyby więc w teście Turinga sędzią uczynić doskonałego telepatę, czy maszyna byłaby w stanie przejść ów test?
Zdaniem Turinga jest to jak najbardziej możliwe — przecież jeżeli uznajemy możliwość pozazmysłowej percepcji, *wszystko* może się zdarzyć.

W dalszej części niniejszego rozdziału poświęcę uwagę głównie bardziej współczesnym rozważaniom dotyczącym TT. Wydaje się, że dyskusje toczące się wokół zagadnienia adekwatności testu Turinga, jako narzędzia służącego badaniu obecności inteligencji w sztucznych systemach poznawczych, można podzielić na dwa nurty (różniące się od siebie kryterium przyjmowanym do oceny TT). W tym celu skorzystam z propozycji R. Frencha przedstawionej w artykule „Subcogniton and the Limits of the Turing Test" [French 1990]. French wyróżnia w nim dwie tezy TT:

1. Tezę o teście Turinga (*the TT Claim*): gra opisywana przez Turinga stanowi dobry test do rozpoznawania obecności inteligencji.
2. Tezę o maszynie myślącej (*the Thinking Machine Claim*): odpowiednio zaprogramowany komputer może zdać test Turinga.

R. French nazywa je odpowiednio *tezą filozoficzną* i *tezą pragmatyczną*. Dyskusje skoncentrowane wokół tezy filozoficznej zmierzają raczej w kierunku wykazania potrzeby porzucenia TT. Wskazuje się tu na rolę behawioryzmu w teście oraz na jego funkcjonalistyczny charakter. Adekwatność TT jest więc postrzegana przez pryzmat *definicji posiadania inteligencji, dostarczanej przez TT*. W dyskusjach dotyczących tezy pragmatycznej możemy wyróżnić dwa skrajne stanowiska. Zwolennicy pierwszego z nich uważają, że test Turinga jest zbyt łatwy, podczas gdy zwolennicy drugiego utrzymują, że TT jest zbyt trudny, aby zdała go (nawet inteligentna) maszyna. Wskazuje się tutaj na pewne rozwiązania, dzięki którym TT stanie się bardziej wiarygodny, lepszy, możliwy do praktycznego zastosowania etc. Adekwatność testu

[1] Pojęcie problemu ramy (*frame problem*) pojawiło się po raz pierwszy w artykule Johna McCarthy'ego i P. Hayesa „Some Philosophical Problems from the Standpoint of Artificial Intelligence" [McCarthy, Hayes 1969]. Problem ramy możemy najogólniej rozumieć jako problem dotyczący tego, które z przekonań podmiotu poznającego powinny być uaktualniane (a które nie) w trakcie interakcji z otoczeniem. Problem ramy posiada zasadniczo dwie wersje — formalną (na gruncie badań nad sztuczną inteligencją) oraz bardziej ogólną (sformułowaną na gruncie dociekań filozoficznych — w ramach epistemologii). Omówienie genezy oraz ewolucji problemu ramy zainteresowany Czytelnik znajdzie m.in. w [Shanahan 2003], [Reiter 2001] oraz we wspomnianej w tekście pozycji autorstwa L. J. Crocketta [Crockett 1994].

wiąże się w tym przypadku z *pragmatycznym aspektem odróżniania człowieka i maszyny* w sytuacji TT.

2.2. Filozoficzna teza testu Turinga

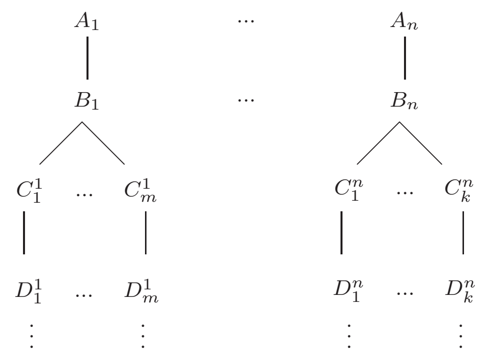
Przedstawię teraz pewien typ argumentacji skierowanej przeciwko testowi Turinga, który można określić mianem argumentu z zupełnego systemu konwersacyjnego (por. [Łupkowski 2006]). Idea tego typu argumentacji sprowadza się do wykazania, że nawet jeśli maszyna zdałaby test Turinga, to i tak nie moglibyśmy powiedzieć o niej, że jest inteligenta, ponieważ TT dostarcza z gruntu błędnej definicji posiadania inteligencji przez maszyny. Jak ujmuje to N. Block:

> [...] problem z testem Turinga rozpatrywanym z perspektywy teoretycznej jest taki, że duży nacisk kładzie się w nim raczej na zachowanie niż na kompetencję. Oczywiście zachowanie jest wskaźnikiem posiadania kompetencji, ale sedno naszego rozumienia umysłu leży w kompetencjach umysłowych, a nie obserwowalnym zachowaniu [Block 1995*b*, s. 384].

Jako najbardziej reprezentatywne sformułowania tego typu argumentacji wybrałem argument z drzewa konwersacji autorstwa N. Blocka (wraz z jego historycznie wcześniejszym ujęciem autorstwa S. Lema) oraz argument chińskiego pokoju przedstawiony przez J. Searle'a.

2.2.1. Argument z drzewa konwersacji

Dokładne sformułowanie argumentu Blocka znajduje się w artykule „The Mind as the Software of the Brain" [Block 1995*b*]. Dla uproszczenia, Block nakłada górną granicę na czas trwania testu Turinga, która wynosi godzinę. Przy takim założeniu możliwe jest — zdaniem Blocka — spisanie drzewa konwersacji, zawierającego wszystkie możliwe jej warianty, które mogłyby wystąpić w ciągu jednej godziny. Block zakłada również, że mogłaby istnieć maszyna, której możliwości pamięciowe pozwoliłyby na przechowywanie takiej struktury. Jak miałoby powstać wspomniane drzewo wypowiedzi? W pierwszym kroku programiści powinni spisać wszystkie, dające się zapisać, ciągi znaków (przy czym powinny to być takie ciągi znaków, które dają się wygenerować w ciągu jednej godziny). Możemy oznaczyć je jako $A_1, ..., A_n$. W kolejnym kroku programiści powinni wyszukać wszystkie możliwe i adekwatne reakcje werbalne dla każdego A i ułożyć je w listę $B_1, ..., B_n$ (gdzie B_1 byłoby odpowiedzią na A_1). Następnie należałoby wybrać odpowiedzi na każde B i uporządkować je w listę oznaczoną literami C z odpowiednim indeksem. Procedura taka powinna być wykonywana aż do uzyskania wszystkich możliwych przebiegów konwersacji dla trwającego jedną godzinę testu Turinga. Ostatecznie więc uzyskujemy strukturę przedstawioną na rysunku 2.1.



Rysunek 2.1. Drzewo konwersacji [Block 1995*b*, s. 382] (opis w tekście)

Poszczególne ścieżki w tym drzewie (np. $A_1 - B_1 - C_1^1 - D_1^1 - ...$) przedstawiają możliwy przebieg konwersacji. Każda z tych konwersacji rozpoczyna się jednym z ciągów znaków należących do listy A, np. A_i, po którym następuje odpowiedź maszyny (znajdująca się na liście B) — B_i i tak dalej. Ned Block opisuje działanie maszyny korzystającej z drzewa konwersacji w sposób następujący:

> Maszyna działa w sposób następujący: rozpoczyna sędzia, cokolwiek wpisze należało będzie do jednego z $A_1, ..., A_n$. Maszyna lokalizuje owo A (niech to będzie powiedzmy A_{2398}), po czym w odpowiedzi podaje B_{2398}, czyli adekwatną odpowiedź na A_{2398} wybraną wcześniej przez programistów. Sędzia wpisuje kolejną wiadomość, a maszyna odnajduje ją na liście C, która znajduje się poniżej B_{2398} w drzewie wypowiedzi i ponownie odpowiada jedną z przygotowanych wcześniej odpowiedzi (która uwzględnia to, co było powiedziane wcześniej w A_{2398} i B_{2398}). I tak dalej [Block 1995*b*, s. 383].

Zdaniem Blocka maszyna tego typu mogłaby z łatwością zdać test Turinga. Znając jednak dokładnie zasady jej działania, nie moglibyśmy przypisać jej posiadania inteligencji. Wiemy bowiem, że wykorzystuje ona jedynie dane ze swojej pamięci, nie podejmując żadnych procesów rozumowania. Pozostaje to w sprzeczności z naszymi najbardziej podstawowymi przekonaniami dotyczącymi tego, czym jest inteligencja.

Argument tego samego typu skonstruował również Stanisław Lem. Przedstawił go w *Summa technologiae*.

> Zauważmy ze swej strony, że grę można skomplikować. Mianowicie są do pomyślenia dwa rodzaje maszyn. Pierwsza jest „zwykłą” maszyną cyfrową, która jest złożona jak mózg ludzki; można z nią grać w szachy, rozmawiać o książkach, o świecie, na wszelkie w ogóle tematy. Gdybyśmy ją otworzyli, ujrzelibyśmy ogromną ilość obwodów sprzężonych tak, jak są sprzężone obwody neuronów w mózgu, poza tym — jej bloki pamięci itd., itp.
>
> Druga maszyna jest zupełnie inna. Jest to do planety (albo do kosmosu) powiększony Gramofon. Posiada ona bardzo dużo, np. sto trylionów nagranych odpowiedzi na wszelkie możliwe pytania. Tak więc, gdy pytamy, maszyna wcale niczego „nie rozumie”, a tylko forma pytania, tj. kolejność drgań naszego głosu, uruchamia przekaźnik, który puszcza w obroty płytę czy taśmę z nagraną odpowiedzią. Mniejsza o stronę techniczną. Rozumie się, że maszyna taka jest nieekonomiczna, że jej nikt nie zbuduje, bo i to właściwie

> niemożliwe, i głównie, nie wiadomo po co by to robić. Ale nas interesuje strona teoretyczna. Bo jeśli o tym, czy maszyna ma świadomość, decyduje zachowanie, a nie budowa wewnętrzna, czyż nie dojdziemy pochopnie do wniosku, że „kosmiczny gramofon" ją posiada — i tym samym wypowiemy nonsens? (A raczej nieprawdę) [Lem 1996*a*, s. 167].

Łatwo zauważyć, że argument autorstwa N. Blocka zawiera znacznie więcej szczegółów, ale wydaje się, że podstawowe intuicje oraz cele obu autorów były w gruncie rzeczy takie same. Hipotetyczna maszyna zaprojektowana przez Blocka przechowuje nie tylko odpowiedzi na pytania, ale również odpowiedzi na komentarze sędziego itp. Jednakże w *Summa technologiae* Lem przyznaje również, że „kosmiczny gramofon", który nazwalibyśmy wystarczająco wyrafinowanym, przechowywałby „nie tylko poszczególne odpowiedzi na możliwe pytania, ale całe sekwencje rozmów, jakie tylko mogą być prowadzone" [Lem 1996*a*, s. 167]. Doskonale oddaje to intuicje związane z budowaniem drzewa konwersacji w przypadku argumentu N. Blocka[2].

2.2.2. Johna Searle'a argument chińskiego pokoju

Kolejnym, bardzo znanym, sformułowaniem argumentu nazwanego tutaj argumentem z zupełnego systemu konwersacyjnego, jest argument chińskiego pokoju autorstwa J. Searle'a. Przez wielu autorów to właśnie ten argument uważany jest za najbardziej trafną krytykę testu Turinga (por. np. [Casacuberta 2007, s. 69] oraz [Searle 2009]). Jest to o tyle niezrozumiałe, że jego wymowa wydaje się — w kontekście rozważań nad TT — dokładnie taka sama, jak omówione powyżej argumenty z drzewa konwersacji i kosmicznego gramofonu.

Argument, znany pod nazwą chińskiego pokoju, przedstawił Searle po raz pierwszy w artykule pod tytułem: „Umysły, mózgi i programy", który ukazał się w piśmie *Behavioral and Brain Sciences* [Searle 1980]. Obszernie tym problemem zajął się także w swojej późniejszej książce poświęconej tematyce sztucznej inteligencji i problemowi umysł-ciało, pod tytułem: *Umysł, mózg i nauka* [Searle 1995].

Argumentacja Searle'a ma na celu wykazanie, że tzw. silna teza o sztucznej inteligencji (*strong* AI) jest błędna[3]. Zwolennicy tej tezy akceptują — zdaniem Searle'a — dwa założenia:

1. Proces myślenia może być w pełni opisany przez algorytmy.

---

[2] Zadziwiające podobieństwo obu argumentów zasługuje na małą uwagę bibliograficzną. Pierwsze sformułowanie argumentu N. Blocka znajdziemy w jego pracy *Troubles with functionalism* (por. [Block 1995*a*]) wydanej w 1978 r. Stanisław Lem sformułował swój argument już w 1963 r. (por. [Lem 1999, s. 198]). Obydwa argumenty powstały oczywiście niezależnie. Warto jednak podkreślić historyczne pierwszeństwo na tym polu Stanisława Lema, ponieważ w literaturze powszechnie za „ojca" tego typu argumentacji uznaje się Neda Blocka (por. [Łupkowski 2006]).

[3] Warto zauważyć, że wprowadzone przez J. Searle'a rozróżnienie na słabą i silną tezę o sztucznej inteligencji nie jest bezdyskusyjnie przyjmowane w literaturze przedmiotu — por. np. dyskusję na ten temat w [Münch 1990].

2. Algorytmy można implementować w pewnego rodzaju maszynie automatycznej — komputerze. Odpowiednio skomplikowany i wyrafinowany algorytm wytworzy świadomość.

2.2.2.1. Program R. Schanka

Przesłanką, która doprowadziła do sformułowania argumentu chińskiego pokoju, był program badawczy Rogera Schanka (z Uniwersytetu Yale). Chociaż Searle podkreśla, że jego argumentacja odnosi się do każdego rodzaju maszyny cyfrowej, to właśnie program Schanka odcisnął szczególne piętno na Searle'owskim eksperymencie myślowym.

Aby uniknąć, problematycznego skądinąd, definiowania inteligencji, Roger Schank postanowił położyć nacisk na jeden z jej przejawów, który wydał mu się szczególnie charakterystyczny (podobnie jak zrobił to Alan Turing). Zdaniem Schanka tym, co charakteryzuje ludzką inteligencję — i tylko ludzką inteligencję — jest zdolność do rozumienia przekazywanych informacji oraz wyciągania z nich wniosków. Zdolność ta uwidacznia się podczas odpowiadania na pytania dotyczące wcześniej usłyszanej historii (zwłaszcza jeśli informacja, o którą pytamy, nie została podana w tej historii *explicite*). Schank (wraz ze współpracownikami) stworzył więc program, który z powodzeniem potrafił odpowiadać na pytania dotyczące prostej historii opowiadającej o zamawianiu hamburgera w restauracji. Zdaniem Searle'a, zwolennik silnej tezy o sztucznej inteligencji wyciągnąłby dwa wnioski z pracy Schanka:

(i) Maszyna (z programem Schanka) potrafi odpowiadać na pytania dotyczące przedstawianej jej historii, ponieważ ją *rozumie*.
(ii) Algorytmy zaimplementowane w programie Schanka wiernie odzwierciedlają odpowiednie procedury, z których korzystają ludzie. Tym samym możliwe jest zastosowanie tych algorytmów dla wyjaśnienia tego zakresu ludzkich zachowań, który wydał się Schankowi interesujący.

2.2.2.2. Chiński pokój

Searle nie godzi się na żaden z obu przedstawionych powyżej wniosków. Proponuje rozważenie następującego problemu: załóżmy, że umysł nasz działa tak, jak przedstawiają to zwolennicy silnej tezy o sztucznej inteligencji i zbadajmy konsekwencje tego założenia. A oto eksperyment myślowy przedstawiony przez Searle'a:

> Wyobraźmy sobie [...], że ktoś z nas jest zamknięty w pokoju i że w pokoju tym jest szereg koszy wypełnionych znakami z języka chińskiego. Załóżmy, że osoba ta, podobnie jak autor myślowego eksperymentu, nie zna chińskiego, otrzymała jednak napisaną w jej ojczystym języku książkę reguł manipulowania znakami języka chińskiego. Reguły te opisują używanie symboli w sposób czysto formalny, opisują manipulowanie nimi w sposób syntaktyczny, nie semantyczny. Mogą mieć postać: „Wybierz ten znak z podwójnym zakrętasem z kosza numer jeden i połóż go za znakiem z dwoma zawijasami z kosza numer dwa” [Searle 1995, s. 28–29].

Dalej Searle pisze:

> Przyjmijmy teraz, że w pokoju pojawiają się jakieś nowe symbole, a osoba w nim siedząca otrzymuje instrukcje, jakie chińskie symbole ma wysłać z pokoju w odpowiedzi na te, które się pojawiły. Załóżmy, że siedzący w pokoju nie wie, iż wysyłane przez ludzi z zewnątrz do pokoju symbole nazywane są przez nich pytaniami, zaś symbole, które siedzący w pokoju wysyła na zewnątrz, nazywane są odpowiedziami na pytania. Przyjmijmy poza tym, że programiści napisali na tyle dobry program, zaś siedząca osoba jest do tego stopnia dobra w manipulowaniu symbolami, że jej odpowiedzi są nieodróżnialne od odpowiedzi osoby faktycznie znającej język chiński. Zatem jakaś osoba zamknięta jest w pokoju, w którym wybiera symbole chińskie i wysyła je w odpowiedzi na inne pojawiające się w pokoju chińskie symbole. W sytuacji, jaką tu opisałem, nie ma możliwości, by w wyniku takiej manipulacji formalnie zdefiniowanymi symbolami nauczyć się języka chińskiego [Searle 1995, s. 28–29].

Opisany powyżej eksperyment myślowy ma, zdaniem Searle'a, pokazywać, że oba wnioski zwolenników silnej tezy o sztucznej inteligencji (opierające się na badaniach Schanka) są błędne. Człowiek w chińskim pokoju *nie rozumie* ani słowa z podawanych mu historii, podobnie jak *nie rozumie* zadawanych pytań i dostarczanych przez siebie odpowiedzi. To co robi osoba w chińskim pokoju (a więc i procesor komputera cyfrowego), to jedynie manipulacja symbolami formalnymi (pozbawionymi na tym poziomie jakiegokolwiek sensu). Interpretacja dokonywana jest dopiero przez człowieka, który wprowadza dane do pokoju i jest ich odbiorcą. Jeśli więc uznamy, że programy komputerowe opisują czysto syntaktyczne operacje na symbolach, to algorytmy, które te programy implementują, nie mogą stanowić jedynego źródła zrozumienia ludzkich stanów mentalnych.

> Meritum mojego myślowego eksperymentu jest następujące: realizując taki formalny komputerowy program, z punktu widzenia obserwatora z zewnątrz, zachowujemy się dokładnie tak, jak byśmy rozumieli język chiński, jednocześnie jednak nie znamy ani jednego słowa języka naturalnego [Searle 1995, s. 29].

Jak się wydaje, wymowa argumentu chińskiego pokoju w odniesieniu do testu Turinga jest analogiczna do wymowy zarzutów formułowanych przez S. Lema i N. Blocka. System opisywany w omówionym eksperymencie myślowym jest w stanie pozytywnie przejść TT, lecz — zdaniem Searle'a — nie świadczy to o obecności inteligencji w takim systemie.

2.2.3. Dyskusja

Przedstawione powyżej sformułowania argumentu z zupełnego systemu konwersacyjnego napotykają na pewne problemy w kontekście rozważań o TT. W literaturze przedmiotu można znaleźć imponującą liczbę zarzutów skierowanych przeciwko konkretnym sformułowaniom interesującego nas argumentu (por. m.in. [Münch 1990], [Churchland, Churchland 1991], [Chalmers 1992], [Crockett 1994], [Kloch 1996], [Lem 1996*b*], [Hauser 1997], [Gregory 2000], [Saygin et al. 2001], [Harnish 2002], [Damper 2006], [Hutchens 2009]), dlatego też skupię się raczej na problemach, które wydają się wspólne dla tego typu argumentacji.

Jak już zauważyłem (s. 32), argument z zupełnego systemu konwersacyjnego ma w założeniu wykazać nieadekwatność definicji posiadania inteligencji dostarczanej przez TT. Stanisław Lem, Ned Block i John Searle przedstawiają hipotetyczny system, który pomyślnie przechodzi TT, ale co do którego znamy pewne fakty, dotyczące wewnętrznych mechanizmów jego działania, które to fakty nie pozwalają na nazwanie rozważanego systemu posiadającym inteligencję. Jednakże — w świetle rozważań dotyczących tego, czy test Turinga rzeczywiście ma dostarczać definicji posiadania inteligencji (por. rozdział 1.3) — przytoczone argumenty wydają się wymierzone nieco obok TT (czy też może raczej powiedzieć, w pewne wyobrażenie o teście Turinga). Test Turinga nie ma dostarczać warunków koniecznych i wystarczających dla stwierdzenia istnienia inteligencji w systemach sztucznych. Stanowi jedynie pewną propozycję testu, którego pomyślne przejście wskazuje, że maszyna posiada pewne interesujące nas własności. Przypomnijmy tutaj cytowaną już wcześniej wypowiedź Turinga:

> *Nie chcę podawać definicji myślenia*, ale jeśli bym musiał, to prawdopodobnie nie byłbym w stanie powiedzieć nic ponad to, że jest to coś w rodzaju brzęczenia (*buzzing*), które zachodzi w mojej głowie. Nie sądzę jednak abyśmy musieli w ogóle zgadzać się co do jakiejkolwiek definicji. Najważniejszą sprawą jest aby spróbować wytyczyć linię oddzielającą właściwości mózgu człowieka, o których chcemy dyskutować od tych, które nas nie interesują. [...] Chciałbym zaproponować pewien test, który można by zastosować do maszyn. Można by nazwać go testem sprawdzającym, czy maszyna myśli. Ale lepiej byłoby uniknąć niepotrzebnych dyskusji i powiedzieć, że *maszyny, które pomyślnie przechodzą ów test są (powiedzmy) maszynami Klasy A* [Newman et al. 1952, s. 3–4]; podkreślenia P.Ł.

W świetle tych słów i rekonstrukcji oryginalnych założeń testu Turinga, przyjęcie tego, że TT ma dostarczać abdukcyjnego wyjaśnienia fenomenu posiadania inteligencji wydaje się najbardziej zgodne z intencjami A. M. Turinga (por. rozdział 1). Jeśli zgodzimy się na taką interpretację propozycji Turinga (a wiele argumentów za takim krokiem przemawia), dyskusje wokół filozoficznej tezy TT powinny nieco zmienić swoją optykę i nierozerwalnie łączyć się z rozważaniem praktycznego aspektu testu oraz budowaniem nowych propozycji, które mogłyby TT zastąpić. Aktualna sytuacja została trafnie skomentowana przez G. Piccininiego:

> Autorami znacznej większości literatury poświęconej Turingowi są logicy lub filozofowie, którzy zainteresowani są raczej aktualnymi problemami filozoficznymi, niż pomysłami samego Turinga [Piccinini 2003, s. 24].

W tym kontekście powstaje również pytanie o to, czy przy obecnym stanie badań nad umysłem i inteligencją dysponujemy lepszym rozwiązaniem niż TT, albo też, ujmując zagadnienie bardziej ogólnie, niż testami wejścia/wyjścia (jak nazywa je Larry J. Crockett; por. [Crockett 1994]). Ned Block pisze — w przytoczonym już wcześniej fragmencie „The Mind as the Software of the Brain" (por. [Block 1995*b*, s. 384]) — że koncentrowanie się TT na zachowaniu jest wadą propozycji Turinga. To, na czym powinniśmy się skupić, to badanie kompetencji. Przyznaje jednocześnie, że zachowanie jest jednym z przejawów owej kompetencji. Tym samym, w pewnym sensie, przeczy sam sobie (zwłaszcza jeśli testu Turinga nie będziemy rozumieli

jako definicji posiadania inteligencji). Siła propozycji Turinga leży w tym, że zaoferował on proste kryterium, które w jego czasach (można zaryzykować twierdzenie, że również w obecnych) umożliwiało praktyczne podejście do problemu badania inteligencji systemów sztucznych. J. Crockett twierdzi, że testy wejścia/wyjścia (opierające się na traktowaniu badanych układów jako czarnych skrzynek) mogą pozostać adekwatne nawet jeśli doczekamy się programu, który będzie modelem działania ludzkiego umysłu. Opiera się w tej kwestii na znanym tekście Ch. Cherniaka „Undebuggability and Cognitive Science" [Cherniak 1988]. Cherniak zauważa w nim, że powstanie programu modelującego działanie ludzkiego umysłu jest nieodłącznie związane z tym, że nie będziemy w stanie zrozumieć kodu owego programu jako całości. Przedsięwzięcie zmierzające do zbudowania takiego programu musiałoby być tak ogromne, że jego końcowy wynik byłby nie do ogarnięcia przez jego twórców. Cherniak wysnuwa ten wniosek, analizując wyniki badań nad ludzkim umysłem w zestawieniu z najbardziej wyrafinowanymi programami stworzonymi w ramach SDI (*Strategic Defense Initiative*[4]) — por. [Cherniak 1988, s. 406]. Zdaniem Ch. Cherniaka:

> Obliczeniowe przybliżenie ludzkiego umysłu byłoby (1) ogromnych rozmiarów, (2) „wielogałęziowe" i zorientowane na podejście holistyczne, (3) uzyskane metodą kolejnych zarysów (tj. wygodne obliczeniowo ale formalnie niepoprawne/niekompletne), (4) poskładane pospiesznie (czyli w dużej mierze stanowiłoby niezbyt eleganckie zestawienie różnych procedur). Program modelujący ludzki umysł okazałby się więc zupełnie niepodobny do znanego nam oprogramowania [...] Tym sposobem program w pełni modelujący ludzki umysł okazuje się być niepoznawalną rzeczą samą w sobie [Cherniak 1988, s. 402].

Obecnie dysponujemy jedynie wyobrażeniami na temat programu modelującego całość działania ludzkiego umysłu[5], jednakże uwagi poczynione przez Ch. Cherniaka wydają się intuicyjnie trafne. Uwagi te wskazują na rolę testów wejścia/wyjścia zarówno dla współczesnych, jak i dla przyszłych programów modelujących ludzki umysł. Również dyskusje wokół pragmatycznej tezy TT i propozycje jego ulepszenia lub zastąpienia (opisane w kolejnym podrozdziale) skupiają się na testowaniu zachowania systemów sztucznych.

Oczywiście TT nie jest jedyną możliwą propozycją i nie posiada gwarancji adekwatności po wsze czasy. Jego atrakcyjność leży jednak w inicjowaniu zarówno teoretycznych dyskusji, jak i praktycznych przedsięwzięć, począwszy od konkursu Loebnera, poprzez *Minimum Intelligent Signal Test*, na pomyśle systemów CAPTCHA skończywszy (por. rozdział 4).

---

[4] *Strategic Defense Initiative* był amerykańskim programem tarczy antyrakietowej mającej chronić kraje NATO przed atakiem ze strony Związku Radzieckiego. Pod nazwą SDI program funkcjonował do 1993 roku.

[5] Warto również nadmienić, że nie ma zgody co do metodologii, która pozwoliłaby na uzyskanie takiego programu (por. np. próbę zestawienia paradygmatów, w jakich uprawiana jest dyscyplina sztucznej inteligencji w [Čaplinskas 1998] lub w [Pellen 2009]).

2.3. Pragmatyczna teza testu Turinga

2.3.1. TT jest za trudny dla inteligentnych maszyn

Początkowy optymizm dotyczący zbudowania systemu sztucznego, który mógłby pozytywnie przejść test Turinga, został szybko ostudzony. Okazało się, że — pomimo lokalnych sukcesów — całościowe modelowanie ludzkich kompetencji językowych nastręcza wiele problemów. Zrodziło to ideę ograniczonego TT. Nałożenie restrykcji na TT ma na celu dokładniejsze sformułowanie sytuacji testowej, a tym samym umożliwienie — w ograniczonym zakresie — praktycznego testowania istniejących systemów sztucznych. Najlepiej dziś znaną wersją ograniczonego TT jest konkurs Loebnera.

2.3.1.1. Ograniczony test Turinga (konkurs Loebnera)

Pomysł konkursu sięga 1990 roku, kiedy Hugh Loebner wraz z *The Cambridge Center for Behavioral Studies* rozpisali konkurs na program komputerowy, który najlepiej poradzi sobie w ograniczonym teście Turinga. Na potrzeby konkursu przyjęto dwie zasady, dzięki którym możliwe stało się jego przeprowadzenie. Po pierwsze, *ograniczono tematykę rozmów*. Programiści startujący w konkursie znają tematykę dla danej edycji, a pytający mają obowiązek pozostawania w jej zakresie. Drugie z ograniczeń jest nałożone właśnie na pytającego. Ma się on zachowywać w czasie rozmowy *tak jakby zachowywał się w czasie normalnej codziennej konwersacji* (oczywiście dotyczącej przyjętego wcześniej tematu). Oznacza to mniej więcej tyle, że nie może stosować udziwnionych wypowiedzi, trików czy przemyślnych prowokacji.

Tematami rozmów wybieranymi do konkursu — który odbywa się nieprzerwanie od 1991 r. — były, między innymi: pogawędki na przyjęciu, rozmowy na temat win burgundzkich, niepowodzenia w związkach, dzieła Szekspira, moda damska (por. [Shieber 1994, s. 4]). W ostatnich edycjach konkursu ograniczenie tematyki rozmów zostało w zasadzie zniesione. W 2006 roku program musiał rozpocząć rozmowę od słów: „Hello, my name is John and I am a man" lub „Hello, my name is Joan, and I am a woman"[6]. Z kolei w edycji z 2008 istniał już tylko wymóg, aby program wyraźnie określił swoją płeć.

Ideą konkursu jest, aby każdego roku wzrastały jego trudność oraz stopień wyrafinowania programów biorących w nim udział. Dąży się również do ustalenia jednolitego protokołu przeprowadzania konkursu, tak aby wyniki kolejnych jego edycji były jak najbardziej porównywalne. Proces ten ma w konsekwencji doprowadzić do powstania programu, który zda klasyczny TT (por. [Loebner 2009], [Copple 2009]). Czy tak rzeczywiście się stanie, pozostaje pytaniem otwartym. Właśnie owo pytanie jest najbardziej znaczące w kontekście ograniczonego TT. Pozytywna odpowiedź na nie uzasadnia bo-

[6] Zasady obowiązujące w kolejnych edycjach konkursu Loebnera oraz listę ich zwycięzców zainteresowany Czytelnik może znaleźć na stronie interentowej projektu: http://loebner.net/Prizef/loebner-prize.html.

wiem ideę ograniczonego testu Turinga traktowanego jako dobry punkt wyjścia dla programów, które (w przyszłości) będą mogły zdać klasyczny TT. Zdaniem krytyków ograniczonego TT — takich jak np. S. Shieber — stanowi on jedynie konkurs na program najlepiej posługujący się sztuczkami, które pozwalają zwodzić sędziów (por. też [Mauldin 1994]). Nie przybliża nas jednak wcale do celu, jakim ma być zdanie klasycznego TT przez maszyny. S. Shieber w „Lessons From a Restricted Turing Test" pisze:

> [...] trudno wyobrazić sobie czysto naukowy cel jaki mógłby przyświecać konkursowi Loebnera. Test Turinga z kolei — w swoim oryginalnym ujęciu — miał swój jasny cel: dostarczyć naukowego kryterium dla zademonstrowania, że stworzony przez człowieka artefakt wykazuje inteligentne zachowania. Nawet ten cel zagubiony został w konkursie Loebnera [Shieber 1994, s. 12].

Ta mocna krytyka Shiebera, skierowana przeciwko idei konkursu Loebnera, opiera się na analogii z konkursem Kremera na pierwsze urządzenie latające napędzane siłą mięśni (*human powered flight*). Nagrodę Kremera, ustanowioną w roku 1959, uzyskał zespół Paula Macready'ego w roku 1977 (za projekt *Gossamer Condor*). Zdaniem Shiebera ów konkurs — w przeciwieństwie do konkursu Loebnera — jest przykładem dobrze sformułowanego i ogłoszonego w odpowiednim czasie. Sukces konkursu Kremera uwarunkowany był dwoma czynnikami:

1. Cele konkursu były jasno sformułowane, a w czasie, kiedy go ustanowiono, nie prowadzono badań nad urządzeniami latającymi napędzanymi siłą mięśni. Cel konkursu był więc dobrze określony — doprowadzić do rozwoju badań w tej dziedzinie.
2. Dyscypliny naukowe leżące u podstaw konstrukcji takich urządzeń (aerodynamika, mechanika, anatomia, fizjologia, materiałoznawstwo) były dobrze rozwinięte i ugruntowane.

Zdaniem Shiebera o tych czynnikach nie możemy mówić w kontekście konkursu Loebnera, co sprawia, że z naukowego punktu widzenia nie jest on w stanie spełnić pokładanych w nim nadziei.

Inny zarzut, sformułowany przez S. Zdenka, dotyczy ograniczeń narzucanych na TT przez zasady konkursu Loebnera (por. [Zdenek 2001]). Jego zdaniem wymagania narzucane przez organizatorów konkursu Loebnera sprawiają, że rozmowy testowe mają sprowadzać się jedynie do wymiany informacji (sędziowie nie powinni zadawać żadnych podchwytliwych czy obraźliwych pytań, powinni trzymać się z góry ustalonego tematu), przy czym najlepiej gdyby dotyczyły one obojętnych faktów. Ze względu na charakter owych rozmów, zasady konkursu Loebnera ignorują — zdaniem Zdenka — fakt ścisłego związku pomiędzy językiem a kontekstem społecznym (szczególnie w zakresie, w jakim użycia języka warunkowane są przez kompetencje komunikacyjne).

Warto nadmienić, że choć zagadnienie to — zdaniem S. Zdenka — jest ignorowane w kontekście konkursu Loebnera, to zostało ono zauważone (i docenione) przez producentów komercyjnego oprogramowania z zakresu NLP (*Natural Language Processing*). Dobrym tego przykładem są badania nad społecznymi interakcjami *chatterbotów* prowadzone np. w ramach tzw. *involvement framework* (por. m.in. [De Angeli et al. 1999], [Dryer 1999],

[De Angeli, Lynch, Johnson 2001], [De Angeli, Graham, Johnson, Coventry 2001], [Gratch, Marsella 2005] oraz [van Vugt et al. 2007]).

2.3.1.2. TT a pytania subkognitywne

R. French w artykule „Subcogniton and the Limits of the Turing Test" [French 1990] również stwierdza, że TT jest za trudny. Uważa go wręcz za bezużyteczny jako test posiadania inteligencji przez systemy sztuczne, a to z tego powodu, iż nigdy nie będą one w stanie go zdać. Być może nawet filozoficzna teza TT jest słuszna, ale teza pragmatyczna jest zupełnie błędna.

Uzasadnieniem takiego stanu rzeczy mają być dwa fakty:

1. TT jest zorientowany kulturowo i antropomorficznie — jest testem ludzkiej inteligencji.
2. Poznawczy (kognitywny) oraz fizyczny poziom inteligencji są nierozłączne.

Zdaniem Frencha możliwe jest zaprojektowanie serii tzw. pytań subkognitywnych (*subcognitive questions*), to znaczy pytań, które odsłaniają niskopoziomowe struktury poznawcze (*low-level cognitive structures*). Pomysł tego rodzaju pytań opiera się na badaniach z dziedziny psychologii poznawczej, w szczególności zaś badań nad prymowaniem[7] (*associative priming*). Zauważono np., że jeżeli zaprezentujemy badanym pewien zestaw słów, to znacznie szybciej są oni w stanie rozpoznać słowa, które poprzedzono słowem w pewien sposób z nim związanym. I tak np. słowo „sól" będzie rozpoznane szybciej, jeżeli poprzedzone będzie słowem „pieprz" niż, powiedzmy, słowem „but" czy też ciągiem znaków „gloff". Ludzie zdają się więc dysponować swego rodzaju sieciami połączeń pomiędzy znaczeniami słów, które tworzą się na drodze codziennych doświadczeń. Te sieci połączeń wydają się również w dużej mierze nieuświadamiane. Pytania subkognitywne mają się odwoływać do zależności zawartych w takich właśnie sieciach. Zdaniem Frencha każdy odpowiednio rozbudowany zbiór pytań zawiera pytania tego typu. Aby maszyna poradziła sobie z pytaniami subkognitywnymi, powinna doświadczać świata w podobny do ludzkiego sposób i w zbliżony sposób zbierać doświadczenia na jego temat. Nie pomoże tutaj ograniczanie TT w sposób, w jaki czyni się to w konkursie Loebnera:

> Z całą pewnością, nie chcielibyśmy ograniczać testu Turinga do pytań typu: „Jakie miasto jest stolicą Francji?" lub „Ile boków ma trójkąt?". Jeśli zgodzimy się, że inteligencja w ogóle musi mieć *coś* wspólnego ze zdolnością kategoryzacji, dostrzegania analogii itp., chcielibyśmy zadawać pytania, które mogłyby testować te zdolności. Lecz właśnie owe pytania umożliwiają nam, w niezawodny sposób, zdemaskowanie komputera [French 1990, s. 63].

---

[7] „Poprzedzanie [prymowanie lub torowanie — przyp. P.Ł.] (*priming*) — zjawisko modyfikacji reakcji na bodziec w wyniku uprzedniego działania innego bodźca (prymy), w jakiś sposób powiązanego z bodźcem docelowym" [Nęcka et al. 2006, s. 646] (por. też [Strelau 2000, s. 795]).

French proponuje, aby pytania subkognitywne wykorzystać w teście Turinga za pomocą „gry w ocenianie” (*rating game*). Gra taka polega na tym, że sędzia prezentuje graczowi pary obiektów. Zadaniem gracza jest ocena poziomu dopasowania tych obiektów do siebie. Np. „Oceń na skali 0–10, jak torebka nadaje się na broń”, lub „Oceń, jak »Flugblogs« nadaje się na nazwę misia przytulanki”. Zdaniem Frencha zadania tego typu odwołują się do wspomnianych niskopoziomowych struktur poznawczych i umożliwią sędziemu łatwe zdemaskowanie maszyny w teście Turinga.

Kontrargumentem dla mocnej tezy zaproponowanej przez R. Frencha jest algorytm PMI-IR, autorstwa Petera D. Turneya, przedstawiony w pracy pod wymownym tytułem „Answering subcognitive Turing test questions: A reply to French” [Turney 2001]. Program implementujący ten algorytm wykorzystuje informacje statystyczne uzyskane z dużej kolekcji tekstów (zaczerpniętych z Internetu). Badania Turneya wykazują, że jego program radzi sobie z pytaniami subkognitywnymi zaproponowanymi przez Frencha w stopniu porównywalnym do ludzkiego. Przykładowym zadaniem, na którym Turney testował swój program było, między innymi: *oceń (w skali 1–10) przydatność skórki od banana, łupin orzechów kokosowych i radia jako instrumentów muzycznych*. Program zwrócił następujące wyniki: *skórka od banana*: 1, *łupiny orzechów kokosowych*: 10, *radio*: 4. Wyniki te wpisują się w schemat odpowiedzi udzielanych przez ludzi, którzy najniżej oceniali przydatność skórki od banana, najwyżej zaś przydatność łupin orzechów kokosowych (radio uzyskiwało wyniki pośrednie). Dodatkowo metoda statystyczna zastosowana w programie umożliwia mu wyszukiwanie i rozpoznawanie synonimów słów (tutaj program Turneya radzi sobie podobnie jak człowiek zdający egzamin językowy TOEFL).

2.3.2. TT jest za mało restrykcyjny

Opinia, zgodnie z którą TT jest za trudny, występuje w literaturze przedmiotu znacznie rzadziej, niż pogląd przeciwny — że TT jest za łatwy.

2.3.2.1. Całościowy test Turinga

W artykule „Mind, Machines and Turing: the Indistinguishability of Indistinguishables” Steven Harnad zaproponował pewną skalę testów Turinga (por. Harnad [2000]). Skala ta jest pięciostopniowa i ma obrazować poziomy trudności gry w naśladownictwo. Poziomy te to kolejno: **t1**, **T2**, **T3**, **T4** oraz **T5** — ich zestawienie oraz charakterystyki zawiera tabela 2.1.

Poziomem, który — zdaniem Harnada — ma największe znaczenie dla przyszłości testu Turinga jest poziom **T3**, określany częściej jako całościowy test Turinga (*Total Turing Test* — TTT). Poziomy **t1** oraz **T2** są za słabe dla modelowania rzeczywiście ludzkiej inteligencji, ponieważ to, co badają, to jedynie (mniej lub bardziej arbitralnie) wybrane aspekty ludzkich zdolności

Tablica 2.1. Skala TT wg Harnada (na podstawie: [Harnad 2000], [Saygin et al. 2001] oraz [French 2000])

Poziom	Opis
t1	Modele, które oddają jedynie część naszych zdolności poznawczych (poziom reprezentowany obecnie przez SI).
T2	Gra w naśladownictwo taka, jak ją opisał Turing.
T3	Zupełna nieodróżnialność w funkcjach behawioralnych — wygląd fizyczny systemu sztucznego nabiera tutaj znaczenia.
T4	Nieodróżnialność na poziomie mikrofunkcji (nieodróżnialność na poziomie każdego neuronu czy neurotransmitera — przy czym mogą być one wykonane z dowolnego materiału).
T5	Nieodróżnialność na poziomie elektronów (empiryczna), jedyne różnice mogą wystąpić na poziomie użytej do konstruowania różnych systemów GUTE.

poznawczych. Co więcej, poziom **T2** napotyka na problem ugruntowania symboli (*symbol grounding problem*). Zdaniem Harnada znaczenia symboli są — przynajmniej częściowo — wyprowadzane z interakcji ze światem zewnętrznym. Dlatego właśnie Harnad postuluje, aby całościowemu testowi Turinga poddawane były roboty, które posiadają możliwość pełnej interakcji ze środowiskiem, w którym funkcjonują. Dzięki temu możliwe będzie całościowe testowanie ich funkcji behawioralnych. Aby pozytywnie przejść TTT, zachowania i wygląd robota poddawanego testowi musiałyby być nieodróżnialne od takich samych zachowań i wyglądu ludzi. Niestety, Harnad nie precyzuje, co dokładnie należy do „całości funkcji behawioralnych". Tak ogólnie zarysowane kryterium, ze względu na które mielibyśmy porównywać ludzi i roboty, wydaje się zupełnie bezużyteczne.

Warto nadmienić, że Harnad nie odrzuca poziomu **T2** (czyli oryginalnej propozycji Turinga) jako nieadekwatnego, ale wskazuje na fakt, że każdy robot, który zda test poziomu **T3**, zda również test poziomu **T2**. Jeśli chodzi o poziomy **T4** i **T5**, Harnad uważa ich osiągnięcie za w zasadzie utopijne. Jego zdaniem to właśnie poziom **T3** (całościowy test Turinga) udostępnia nam odpowiednie środki, aby rozstrzygnąć, czy dana maszyna posiada inteligencję.

> [...] to właśnie **T3**, nie zaś **T4** czy **T5**, pozostanie ostatecznym arbitrem. Powód ku temu jest prosty (i znowu jest on już zawarty w kryterium nierozróżnialności funkcjonalniej autorstwa Turinga): nie tylko zwyczajni ludzie nie potrafią czytać w umysłach, nie potrafią tego również inżynierowie. Wszyscy są ograniczeni barierą problemu innych umysłów. Funkcja jest jedynym empirycznym kryterium [Harnad 2000, s. 441-442].

2.3.2.2. Rzeczywiście całościowy test Turinga

Wydawałoby się, że już nie można bardziej „utrudnić" TT, ale zdaniem P. Schweizera nawet na poziomie całościowego testu Turinga nadal nie dysponujemy pełną możliwością orzekania o inteligencji porównywalnej do ludzkiej. Aby taką możliwość otrzymać, powinniśmy skorzystać z zaproponowanego

przez niego rzeczywiście całościowego testu Turinga (*Truly Total Turing Test* — TTTT).

Schweizer zgadza się z Harnadem, że przypisujemy posiadanie inteligencji innym ludziom, obserwując ich zachowania (a więc na poziomie analizy funkcjonalnej), ale zauważa, że jest to możliwe tylko dzięki temu, że mamy pewną ogólną wiedzę na temat *typu* podmiotu, o którym chcemy orzekać. Posiadamy coś, co Schweizer nazywa historycznym rejestrem (*historical record*) zdolności kognitywnych podmiotu tego rodzaju, jakim jest człowiek (człowiek stworzył język, narzędzia, gry itp.). P. Schweizer proponuje więc długoterminowe, ewolucyjne kryterium badania inteligencji, podkreślając, że:

> [...] TTTT nie stanowi testu dla indywidualnych systemów poznawczych. Jest on raczej pomyślany jako test możliwości pewnych architektur poznawczych, których przedstawicielami są poszczególne systemy poznawcze. TTTT nie implikuje więc tego, że każda osoba (lub robot) aby zostać uznaną za inteligentną musiałaby [...] dokonać wszystkich przełomowych odkryć w historii ludzkości [Schweizer 1998, s. 267–268].

W tym świetle propozycję Harnada należałoby raczej zakwalifikować do poziomu **t1** (w jego własnej hierarchii). Zdaniem Schweizera, naczelną wadą całościowego testu Turinga jest to, że nakazuje badanie pojedynczych egzemplarzy podmiotów poznawczych operujących na modelach świata zadanych im *a priori*. Zamiast tego należałoby się raczej skupić na badaniu typów podmiotów poznawczych działających w realnym świecie na przestrzeni określonego czasu (por. [Schweizer 1998, s. 267]). Dopiero kiedy podmioty poznawcze (rozpatrywane jako pewien typ architektury poznawczej) przejdą tak rozumiany rzeczywiście całościowy test Turinga, będziemy mogli z powodzeniem stosować tradycyjny test Turinga oraz całościowy test Turinga w odniesieniu do poszczególnych egzemplarzy podmiotów poznawczych[8].

Celem S. Harnada oraz P. Schweizera było ulepszenie testu Turinga. Ulepszenie to dotyczy nie tyle samej konstrukcji TT, ile raczej kryterium posiadania inteligencji, na którym się on opiera. Zdaniem Harnada i Schweizera, kryterium to jest zbyt wąskie, aby rzeczywiście nadawało się do badania sztucznych systemów poznawczych. S. Harnad proponuje więc, aby testować całość funkcji behawioralnych takiego systemu, zaś R. Schweizer posuwa się jeszcze dalej, proponując długoterminowe, ewolucyjnie zorientowane kryterium badania typów architektur poznawczych (a nie pojedynczych ich przedstawicieli). Niewątpliwie obie te propozycje są bardzo interesujące pod względem teoretycznym. Atrakcyjność zarysowanych w nich kryteriów oceny sztucznych systemów poznawczych kryje się w tym, że mają maksymalnie zbliżyć się do kryteriów, z których korzystamy na co dzień, przypisując innym ludziom stany mentalne. Jest to jednocześnie poważna wada całkowitego testu

[8] Modyfikacje testu Turinga w podobnym duchu odnajdziemy np. w artykule „Intelligence is not Enough: On the Socialization of Talking Machines" [Ronald, Sipper 2001], w którym autorzy starają się zidentyfikować te aspekty inteligencji, które nie mogą zostać zanalizowane w oryginalnym teście Turinga. Ciekawa jest również propozycja „długoterminowego testu Turinga" przedstawiona przez B. Edmondsa w tekście „The Constructiblity of artificial intelligence (as defined by Turing test)", [Edmonds 2000]. Z kolei Eugeniusz Szumakowicz używa argumentacji zbliżonej do tej zastosowanej przez P. Schweizera, aby wykazać zupełną nieadekwatność kryterium posiadania inteligencji oferowanego przez test Turinga (por. [Szumakowicz 2000], por. też [Łupkowski 2005*a*]).

Turinga i rzeczywiście całkowitego test Turinga — liczba aspektów, które należałoby brać pod uwagę, przeprowadzając takie testy, praktycznie dyskwalifikuje je jako rzeczywiste narzędzie badawcze. Warto zwrócić uwagę na fakt, że — stosunkowo wąskie — kryterium zaproponowane przez Turinga nie było przypadkowe. Zdaniem Turinga: „Najważniejszą sprawą jest aby spróbować wytyczyć linię oddzielającą właściwości mózgu człowieka, o których chcemy dyskutować od tych, które nas nie interesują" [Newman et al. 1952, s. 3–4]. Dzięki takiemu podejściu możemy sobie wyobrazić praktyczne wykorzystanie testu Turinga (na co wskazuje choćby konkurs Loebnera czy też systemy CAPTCHA opisane w rozdziale 4). W tym kontekście propozycjami znacznie bardziej interesującymi niż TTT i TTTT są odwrócony test Turinga, *Minimum Intelligence Signal Test* oraz test lady Lovelace omówione w rodziale czwartym tej książki.

2.4. Twierdzenie Harraha w kontekście TT

Argument matematyczny jest jednym z najbardziej interesujących rozpatrywanych przez Turinga w „Computing Machinery". Jest on interesujący zarówno w kontekście samego testu Turinga, jak i jako element szeroko zakrojonych dyskusji dotyczących zagadnienia sztucznej inteligencji oraz ograniczeń ludzkiego umysłu.

Za A. M. Turingiem przypomnę sformułowanie tego argumentu:

> Istnieją wyniki na gruncie logiki matematycznej, które mogą zostać wykorzystane dla wykazania pewnych ograniczeń maszyn o stanach dyskretnych. Najbardziej znanym z takich wyników jest twierdzenie Gödla [...] Istnieją również inne — pod wieloma względami podobne — wyniki osiągnięte przez Churcha, Kleene'go, Rossera i Turinga. Szczególnie ostatni z tych wyników jest wart rozważenia w tym miejscu, ponieważ bezpośrednio odnosi się on do maszyn [o stanach dyskretnych] [...]. Wynik ten głosi, że pewnych rzeczy takie maszyny nie są w stanie zrobić. Jeśli taka maszyna musiałaby udzielać odpowiedzi na pytania takie, jak w grze w naśladownictwo, istniałyby pytania, na które udzieliłaby ona błędnych odpowiedzi, lub nie udzieliłaby ich wcale (niezależnie od ilości czasu, jaki miałaby do dyspozycji) [Turing 1950, s. 444].

Wyniki, o których wspomina Turing w przytoczonym powyżej cytacie, określane są mianem twierdzeń limitacyjnych[9]. Warto wspomnieć, że te twierdzenia (zwłaszcza twierdzenie Gödla) są bardzo często wykorzystywane w dyskusjach o naturze ludzkiego umysłu (por. m. in. [Lucas 1961], [Marciszewski 1998], [Woleński 1999], [Penrose 2000], [Hetmański 2000], [Shagrir 2002], [Krajewski 2003]).

Z perspektywy tej pracy szczególnie interesujące są jednak wyniki osiągnięte na gruncie logiki pytań w postaci rozszerzonego twierdzenia Harraha. Wynika to z naturalnego — moim zdaniem — potraktowania testu Turinga jako pewnego systemu pytań i odpowiedzi (por. rozdział 3.1). Logika pytań dostarcza narzędzi użytecznych do zbadania takiego systemu oraz konsekwencji tego, że w teście bierze udział maszyna. Taki krok pozwala również — do

[9] Sformułowania twierdzeń, o których pisze Turing, można znaleźć np. w [Krajewski 2003, s. 63–71] oraz w [Murawski 2000].

pewnego stopnia — uniknąć bardzo problematycznych (i, jak się wydaje, na dzień dzisiejszy nierozstrzygalnych) dyskusji dotyczących algorytmiczności (lub jej braku) ludzkiego umysłu.

Przyjmuję tutaj oryginalne założenia testu Turinga, przy których to sędzia zadaje pytania (oraz ewentualnie komentuje odpowiedzi), zaś gracz udziela jedynie odpowiedzi (nie generuje pytań). Na „wejściu" gracza-maszyny mogą więc pojawić się wypowiedzi będące pytaniami, zdaniami oznajmującymi lub wypowiedzi nie należące do żadnej z tych kategorii (w kontekście testu Turinga interesujące będą pytania i zdania oznajmujące). Na „wyjściu" gracza-maszyny pojawiały się będą jedynie zdania oznajmujące (będące odpowiedziami udzielanymi przez maszynę na pytania sędziego). Odnośnie do zdań oznajmujących zakładam, że maszyna posiada początkowy ich zbiór (można go określić jako początkową bazę wiedzy maszyny). Do tego zbioru — w miarę trwania testu — dodawane są kolejne zdania (np. poprzez dołączanie wyjaśnień udzielanych przez sędziego). Zakładam również, że maszyna dysponuje pewnymi „możliwościami dedukcyjnymi", które dobrze imitują odpowiednie możliwości sędziego.

Zgodnie z oryginalnymi założeniami testu Turinga, maszyna udziela jedynie odpowiedzi, nie formułując pytań. Można przyjąć, że odpowiedzi maszyny będą zdaniami oznajmującymi. Przyjmuję, że maszyna udziela odpowiedzi bezpośrednich w sensie Belnapa. Są to takie odpowiedzi, które „bezpośrednio i precyzyjnie odpowiadają na pytanie, podając dokładnie tyle informacji ile potrzeba" [Belnap 1969, s. 124]. Założenie to — w dużym stopniu — gwarantuje utrzymanie płynności konwersacji w teście Turinga[10] oraz wydaje się spełniać wymaganie nałożone przez Turinga na zachowanie gracza w TT, mówiące, że powinien on udzielać odpowiedzi tak, jakby zrobił to człowiek. Co więcej, przyjmuję, że odpowiedzi bezpośrednie udzielane przez maszynę muszą być dodatkowo *trafne z uwagi na warunki zadania* (wyrażonego pytaniem sędziego).

O graczu-maszynie w TT zakładam również, że wykonywane przez niego procedury muszą być efektywne. Intuicyjne pojęcie procedury efektywnej[11] wyeksplikuję tutaj za pomocą pojęcia funkcji rekurencyjnej, utożsamiając wykonywanie procedury efektywnej z obliczaniem (wartości) funkcji rekurencyjnej. Innymi słowy, przyjmę tutaj, że „aktywność" gracza-maszyny polega na wykonywaniu pewnych algorytmów, będących w istocie obliczaniem określonych funkcji rekurencyjnych (całkowitych lub częściowych).

Jak wiadomo, pojęcie funkcji rekurencyjnej można określać na wiele równoważnych sposobów (por. np. [Murawski 2000], [Hopcroft, Ullman 2003], [Papadimitriou 2002], [Dalen 2002]). Skorzystam z ujęć przedstawionych w [Krajewski 2003] oraz [Ławrow, Maksimowa 2004].

---

[10] Należy pamiętać, że zdaniem Turinga zarówno jedna, jak i druga strona biorąca udział w zaprojektowanym przez niego teście miały zachowywać się tak, jak przy codziennej konwersacji.

[11] Za kluczowe własności procedury efektywnej uważa się skończoną opisywalność i składanie się z kolejnych kroków, z których każdy może być wykonany w sposób mechaniczny — por. [Hopcroft, Ullman 2003, s. 172].

Rozważam funkcje — całkowite lub częściowe — o argumentach i wartościach będących liczbami naturalnymi; mówiąc dalej o liczbach, będę miał na myśli liczby naturalne, natomiast symbolem $\mathbb{N}$ oznaczał będę zbiór liczb naturalnych.

Mianem funkcji wyjściowych określa się: funkcję stałą $Z(x) = 0$, funkcję następnika $S(x) = x + 1$ oraz funkcje rzutowania $I_i^n(x_1, ..., x_n) = x_i$, dla dowolnego $n \in \mathbb{N}$ oraz $i \leq n$.

Mówimy, że funkcja f dana równością:

$$f(x_1, ..., x_n) = g(h_1(x_1, ..., x_n), ..., h_m(x_1, ..., x_n))$$

jest otrzymywana z funkcji g, $h_1, ..., h_m$ poprzez operację składania.

Powiemy, że funkcja $n+1$ argumentowa f jest otrzymywana z n-argumentowej funkcji g oraz $n+2$-argumentowej funkcji h za pomocą operacji rekursji prostej wówczas, gdy f spełnia następujące równości:

$$f(0, x_1, ..., x_n) = g(x_1, ..., x_n)$$

$$f(x+1, x_1, ..., x_n) = h(f(x, x_1, ..., x_n), x, x_1, ..., x_n)$$

Funkcja pierwotnie rekurencyjna to każda funkcja, którą można otrzymać w skończenie wielu krokach z funkcji wyjściowych poprzez zastosowanie operacji składania oraz operacji rekursji prostej.

Mówimy, że funkcja f jest otrzymana z funkcji g poprzez operację minimum wówczas, gdy $f(x_1, ..., x_n)$ jest określone i równe y wtedy i tylko wtedy, gdy $g(x_1, ..., x_n, 0)$, ..., $g(x_1, ..., x_n, y-1)$ są wszystkie określone i różne od 0, a $g(x_1, ..., x_n, y) = 0$.

Funkcja częściowo rekurencyjna to funkcja, którą można otrzymać z funkcji wyjściowych za pomocą skończonej liczby zastosowań operacji składania, operacji rekursji prostej oraz operacji minimum.

Mówiąc dalej o funkcjach rekurencyjnych, będę miał na myśli funkcje częściowo rekurencyjne. Całkowite funkcje (częściowo) rekurencyjne określał będę mianem funkcji ogólnie rekurencyjnych.

Na mocy tezy Churcha-Turinga klasa funkcji obliczalnych jest równa klasie funkcji (częściowo) rekurencyjnych (por. [Hopcroft, Ullman 2003, s. 192], [Murawski 2000, s. 63]).

Mając dane pojęcie funkcji rekurencyjnej, mogę następnie określić pojęcia zbioru rekurencyjnego oraz zbioru rekurencyjnie przeliczalnego (*recursively enumerable*). Aby to zrobić, scharakteryzuję pojęcie funkcji charakterystycznej. Funkcją charakterystyczną zbioru (liczb naturalnych) X nazywamy funkcję: $K_X : \mathbb{N} \to \{1, 0\}$ spełniającą następujący warunek:

$$K_X(x) = \begin{cases} 0, & \text{jeśli } x \in X \\ 1, & \text{jeśli } x \notin X \end{cases}$$

Zbiór $X \subseteq \mathbb{N}$ jest rekurencyjny wtedy i tylko wtedy, gdy funkcja charakterystyczna zbioru X jest ogólnie rekurencyjna.

Pojęcie zbioru rekurencyjnego odnosi się, ściśle rzecz biorąc, do zbiorów liczb naturalnych. Gdy pragniemy je zastosować do zbioru wyrażeń (co bę-

dzie niezbędne w kontekście rozważań poświęconych testowi Turinga), zakładamy, że elementy tego zbioru są kodowane przez liczby naturalne. Zbiór wyrażeń W określamy mianem rekurencyjnego wówczas, gdy zbiór kodów elementów zbioru W jest rekurencyjny. Intuicyjny sens pojęcia rekurencyjnego zbioru wyrażeń jest następujący: zbiór wyrażeń W jest rekurencyjny wtedy i tylko wtedy, gdy istnieje mechaniczna metoda, która dla dowolnego wyrażenia w pozwala w skończonej liczbie z góry przepisanych kroków stwierdzić, czy $w \in W$, czy też $w \notin W$. Mówiąc ogólnie, zbiór wyrażeń jest rekurencyjny wówczas, gdy istnieje efektywna metoda rozstrzygania czy dane, dowolne wyrażenie jest elementem tego zbioru, czy też nie jest.

Mówimy, że relacja $R \subseteq \mathbb{N}^n$ jest rekurencyjna wtedy i tylko wtedy, gdy funkcja charakterystyczna relacji R jest ogólnie rekurencyjna.

Funkcją charakterystyczną relacji $R \subseteq \mathbb{N}^n$ nazywamy funkcję: $K_R : \mathbb{N}^n \to \{1, 0\}$ spełniającą następujący warunek:

$$K_R(x_1, ..., x_n) = \begin{cases} 0, & \text{jeśli } \langle x_1, ..., x_n \rangle \in R \\ 1, & \text{jeśli } \langle x_1, ..., x_n \rangle \notin R \end{cases}$$

Zbiór $X \subseteq \mathbb{N}$ jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy dla pewnej relacji $R \subseteq \mathbb{N}^2$ takiej, że funkcja charakterystyczna relacji R jest ogólnie rekurencyjna zachodzi:

(*) $X = \{x \in \mathbb{N} : \exists y R(x, y)\}$

Można udowodnić (por. [Dalen 2002, s. 278]), że warunek (*) jest równoważny warunkowi:

(**) X jest zbiorem wartości pewnej funkcji (częściowo) rekurencyjnej.

Podobnie jak poprzednio, pojęcie zbioru rekurencyjnie przeliczalnego można — pośrednio — odnieść do zbioru wyrażeń. Intuicyjnie rzecz biorąc, rekurencyjnie przeliczalny zbiór wyrażeń W to taki, dla którego istnieje mechaniczna metoda, która dla dowolnego wyrażenia $w \in W$ pozwala w skończonej liczbie z góry przepisanych kroków stwierdzić, że $w \in W$, natomiast gdy $w \notin W$, to metoda ta może nie dać żadnej odpowiedzi na pytanie „Czy $w \in W$?" Mówiąc ogólnie, zbiór wyrażeń jest rekurencyjnie przeliczalny wówczas, gdy istnieje efektywna metoda, która dla każdego wyrażenia należącego do tego zbioru, pozwala pokazać/rozstrzygnąć, że jest tak właśnie.

Jest oczywiste, że każdy rekurencyjny zbiór wyrażeń jest zarazem rekurencyjnie przeliczalny, jednakże nie każdy zbiór rekurencyjnie przeliczalny jest rekurencyjny.

W celu zbadania konsekwencji twierdzenia Harraha dla testu Turinga posłużę się rozszerzoną wersją tego twierdzenia zaproponowaną w artykule „Interrogatives, Recursion and Incompleteness" autorstwa A. Wiśniewskiego i J. Pogonowskiego [Wiśniewski, Pogonowski 2010]. W pierwszej kolejności przytoczę jednak twierdzenie Harraha w jego oryginalnym sformułowaniu:

Twierdzenie 1. *Niech L będzie językiem takim, że: (i) istnieje nieskończenie wiele wyrażeń języka L, które są uporządkowane alfabetycznie, (ii) pewne*

wyrażenia języka L są zdaniami; zbiór zdań jest rekurencyjny. Załóżmy, że istnieje zbiór pytań S taki, że: (iii) S jest rekurencyjnie przeliczalny, (iv) każde pytanie języka L posiada nieskończenie wiele odpowiedzi bezpośrednich lub można mu przypisać nieskończenie wiele odpowiedzi w sposób neutralny logicznie, (v) dla każdego pytania Q zbiór odpowiedzi bezpośrednich na Q jest rekurencyjnie przeliczalny, (vi) odpowiedzi bezpośrednie na pytania ze zbioru S są zdaniami języka L. Wtedy istnieje zbiór X będący zbiorem zdań języka L taki, że (1) X nie jest zbiorem odpowiedzi bezpośrednich na żadne z pytań ze zbioru S, oraz (2) istnieje rekurencyjna własność P taka, że każdy element zbioru X posiada tę własność (por. [Wiśniewski 1995, s. 98]; por. też [Harrah 1969, s. 160] i [Harrah 2002, s. 10–11]).

W odróżnieniu od oryginalnego twierdzenia Harraha, w jego rozszerzonej wersji przyjmuje się, że w rozważanych językach mogą występować nie tylko pytania nieskończone (czyli takie, które posiadają przeliczalnie nieskończoną ilość odpowiedzi bezpośrednich), ale również pytania skończone. Ponadto teza tego twierdzenia mówi o istnieniu *rekurencyjnych* zbiorów zdań, które nie są zbiorami odpowiedzi na żadne pytanie (por. [Wiśniewski, Pogonowski 2010, s. 4–5]).

Zanim przejdę do treści intersującego mnie twierdzenia, wprowadzę — za autorami wspomnianego artykułu — pojęcia ω-pytania oraz pytania efektywnego (por. [Wiśniewski, Pogonowski 2010, s. 5]). Pod pojęciem *ω-pytania* rozumiemy pytanie, którego zbiór odpowiedzi bezpośrednich jest przeliczalnie nieskończonym zbiorem zdań. Pytanie jest *efektywne* wtedy i tylko wtedy, gdy zbiór jego wszystkich odpowiedzi bezpośrednich jest niepusty i rekurencyjnie przeliczalny.

Interesujące mnie twierdzenie ma następującą postać:

Twierdzenie 2. [Wiśniewski, Pogonowski 2010] *Niech L będzie językiem, takim że: (a) pośród jego wyrażeń znajdują się zdania i pytania, (b) zarówno zdania jak i pytania tego języka mogą zostać zakodowane przy użyciu liczb naturalnych, oraz (c) zbiór zdań języka L jest przeliczalnie nieskończony i rekurencyjny. Jeżeli spełniony jest następujący warunek:*

(*) *każdy (przeliczalnie) nieskończony rekurencyjny zbiór zdań języka L jest zbiorem odpowiedzi bezpośrednich na jakieś pytanie języka L*

to albo zbiór ω-pytań nie jest rekurencyjnie przeliczalny, albo istnieje przynajmniej jedno ω-pytanie języka L, które nie jest efektywne.

W twierdzeniu tym mówi się zatem, że przy spełnieniu założonych warunków, w języku L możemy mieć do czynienia z jedną z dwóch możliwości. W przypadku pierwszej, zbiór ω-pytań nie jest rekurencyjnie przeliczalny, co oznacza tyle, że nie istnieje mechaniczna metoda, która dla każdego pytania należącego do zbioru ω-pytań pozwalałaby rozstrzygnąć, że jest tak właśnie. W przypadku drugim istnieje co najmniej jedno pytanie języka, którego zbiór odpowiedzi bezpośrednich jest przeliczalnie nieskończonym zbiorem zdań, jednakże zbiór ten nie jest rekurencyjnie przeliczalny.

Dowód Twierdzenia 2 zainteresowany Czytelnik znajdzie w przytaczanej już pracy [Wiśniewski, Pogonowski 2010].

Przyjmuję, że język, w którym toczy się dialog maszyny i sędziego, spełnia założenia powyższego twierdzenia. Ponadto przyjmuję, że zbiór ω-pytań tego języka jest rekurencyjnie przeliczalny. Są to założenia silne, ale dzięki nim możliwe jest zagwarantowanie tego, że gracz poddawany testowi będzie dysponował dużymi możliwościami. Spełniający powyższe założenia język, w którym toczy się dialog maszyny i sędziego, będę oznaczał symbolem L^*.

O graczu-maszynie zakładam, co następuje:

1. Dysponuje ona efektywną procedurą P_1 *rozpoznawania, czy* wyrażenie aktualnie dane na „wejściu" jest zdaniem języka L^*, tj. — mówiąc intuicyjnie — procedura P_1 zastosowana do dowolnego zdania języka L^* danego na „wejściu" pozwala maszynie rozpoznać, w skończonej liczbie z góry danych kroków, że jest to zdanie języka L^*, natomiast P_1 zastosowana do każdego danego na „wejściu" wyrażenia, które nie jest zdaniem języka L^*, pozwala maszynie rozpoznać, znów w skończonej liczbie z góry danych kroków, że nie jest to zdanie języka L^*. Oczywiście taka procedura może istnieć tylko wówczas, gdy zbiór zdań języka L^* jest rekurencyjny.
2. Dysponuje ona efektywną procedurą P_2 *rozpoznawania, że* wyrażenie aktualnie dane na „wejściu" i będące ω-pytaniem języka L^*, jest pytaniem języka L^*. I znów, taka procedura może istnieć tylko wówczas, gdy zbiór pytań języka L^* jest rekurencyjnie przeliczalny.
3. Dysponuje ona, dla pewnych pytań języka L^*, *poprawnymi* oraz *efektywnymi* i *zupełnymi* procedurami generowania odpowiedzi bezpośrednich na te pytania. Procedura *poprawna* — to generująca na „wyjściu" zdania faktycznie będące odpowiedziami bezpośrednimi na rozważane pytanie i tylko takie zdania. Procedura *efektywna* — to procedura realizowana, dla każdej odpowiedzi bezpośredniej danej na „wyjściu", w skończonej liczbie z góry danych kroków. Procedura *zupełna* — to procedura generująca, dla każdego zdania będącego odpowiedzią bezpośrednią, to zdanie (jeśli procedura jest zupełna, to mamy gwarancję, że każda odpowiedź bezpośrednia może być wygenerowana). Wyposażenie maszyny w takie procedury jest, rzecz jasna, możliwe tylko wówczas, gdy zbiory odpowiedzi bezpośrednich na pewne pytania języka L^* (ściślej: na te pytania, których dotyczą te procedury) są rekurencyjnie przeliczalne.

Wróćmy teraz do Twierdzenia 2. Ponieważ założyłem, że język, w którym toczy się dialog maszyny i sędziego, to język L^*, na mocy Twierdzenia 2 (uwzględniając odpowiednie charakterystyki dotyczące gracza-maszyny) wnoszę, że istnieje przynajmniej jedno ω-pytanie języka L^*, które nie jest efektywne. Znaczy to, że zbiór odpowiedzi bezpośrednich na to pytanie nie jest zbiorem wartości żadnej funkcji rekurencyjnej, czyli — mówiąc dokładniej — dla każdej funkcji rekurencyjnej f albo istnieje zdanie (ściślej, jego kod, przy ustalonym kodowaniu) będące wartością tej funkcji i nie będące zarazem odpowiedzią bezpośrednią na rozważane pytanie, albo też istnieje zdanie będące odpowiedzią bezpośrednią na analizowane pytanie i nie będące zarazem wartością funkcji f. Wynika stąd, *że dla każdej funkcji rekurencyjnej f, której wartościami są wyłącznie odpowiedzi bezpośrednie na rozważane py-*

tanie istnieje co najmniej jedna odpowiedź bezpośrednia na to pytanie, która nie jest wartością funkcji f. Określmy pytania o tej własności mianem „niedościgłych". Na mocy Twierdzenia 2 co najmniej jedno pytanie języka L^* jest niedościgłe.

Jest oczywiste, że dla pytania niedościgłego nie istnieje procedura generowania odpowiedzi bezpośrednich na to pytanie, która jest zarazem poprawna oraz efektywna i zupełna. Tak więc nie jest możliwe wyposażenie gracza-maszyny, prowadzącego dialog z sędzią w języku spełniającym warunki nakładane na język L^*, w zestaw poprawnych oraz efektywnych i zupełnych procedur generowania odpowiedzi bezpośrednich na wszystkie pytania tego języka. W przypadku pytań „niedościgłych" zaimplementowane procedury mogą być jednak poprawne i efektywne, ale — co należy podkreślić — żadna z nich nie będzie zupełna (nie będziemy mieli więc gwarancji, że każda odpowiedź bezpośrednia może zostać wygenerowana). Co więcej, jest teoretycznie możliwe, że gracz-maszyna będzie wyposażony w wiele poprawnych i efektywnych procedur generowania odpowiedzi bezpośrednich na jakieś pytanie „niedościgłe" i procedury te dają w efekcie różne podzbiory właściwe zbioru odpowiedzi bezpośrednich na to pytanie. Nie można zatem *a priori* powiedzieć, że istnieje jakaś odpowiedź bezpośrednia „absolutnie niedościgła", tj. taka, która nie będzie generowana przez żadną poprawną i efektywną procedurę. Jednakże żaden skończony zestaw procedur tego typu nie da w efekcie procedury zupełnej — albowiem taka dla pytania „niedościgłego" nie istnieje. Co więcej, chociaż można zawsze udoskonalać/rozbudowywać dostępne maszynie procedury, nie istnieje granica takich udoskonaleń, w której maszyna będzie dysponować *skończonym* i *uniwersalnym* zarazem zestawem procedur — tj. dla każdego pytania rozważanego języka co najmniej jedną poprawną, efektywną i zupełną procedurą generowania odpowiedzi bezpośrednich na to pytanie.

Aby wzmocnić pozycję gracza-maszyny w TT można dodatkowo przyjąć następujące założenie:

4. Dla każdego pytania języka L^* gracz-maszyna dysponuje jakąś poprawną i efektywną procedurą generowania odpowiedzi bezpośrednich na to pytanie.

Powyższe założenie wyklucza sytuację, w której gracz-maszyna nie potrafi wygenerować żadnej odpowiedzi bezpośredniej na jakieś pytanie.

Jakie są konsekwencje przyjętych założeń oraz ustaleń poczynionych na bazie Twierdzenia 2 dla zagadnienia adekwatności TT?

Należy pamiętać, że „odpowiedź bezpośrednia" to nie to samo, co „odpowiedź trafna z uwagi na warunki zadania". W TT oczekujemy, że gracz poddawany testowi powinien *udzielać tych spośród odpowiedzi bezpośrednich na zadawane mu pytania, które są trafne z uwagi na warunki stawianych przed nim zadań.* Aby udzielić odpowiedzi bezpośredniej, która spełnia ten warunek, gracz nie musi uprzednio wygenerować wszystkich odpowiedzi bezpośrednich na to pytanie — wystarczy, aby był on w stanie wygenerować tę z nich, która jest trafna z uwagi na warunki zadania (i zarazem oczekiwana przez sędziego). Dotyczy to zarówno gracza-maszyny, jak i gracza-człowieka.

Zadanie graczowi-maszynie pytania niedościgłego nie musi prowadzić do tego, że sędzia nie otrzyma na to pytanie oczekiwanej odpowiedzi i tym samym trafnie zidentyfikuje gracza jako maszynę właśnie. Zdarzyć się może, że sędzia zada graczowi pytanie niedościgłe (nawet bez świadomości, że jest ono niedościgłe) i akurat ta odpowiedź bezpośrednia na to pytanie, która jest — z punktu widzenia sędziego i z uwagi na warunki zadania — trafna znajduje się wśród odpowiedzi, które mogą być efektywnie generowane przez maszynę z uwagi na dostępne jej procedury oraz zarazem wyprowadzone przez maszynę z jej „bazy wiedzy". Wtedy sędzia otrzyma oczekiwaną odpowiedź, chociaż zadane przez niego pytanie było „niedościgłe". Jednakże — w świetle poczynionych wyżej ustaleń — nie ma na to gwarancji. Rzecz w tym, że *dla każdej odpowiedzi bezpośredniej na pytanie można tak dobrać warunki zadania, aby właśnie ta odpowiedź była trafna z uwagi na dobrane warunki zadania* — a więc także dla takiej odpowiedzi, która nie może być wygenerowana za pomocą procedur dostępnych graczowi-maszynie (istnienie takich odpowiedzi jest, przypomnijmy, zagwarantowane przez poczynione wyżej założenia i ustalenia). Jednakże w TT to sędzia ustala warunki zadania oraz zadaje pytania. Ta asymetria, wraz z poczynionymi wyżej ustaleniami, zdaje się pociągać następujący wniosek: chociaż jest możliwe, że gracz-maszyna odniesie sukces w TT przeprowadzanym w ustalonym przedziale czasowym, to zawsze możliwe jest takie przedłużenie przeprowadzanego właśnie testu, w którym gracz-maszyna odniesie porażkę, tj. sędzia trafnie zidentyfikuje gracza-maszynę jako maszynę właśnie. Tak więc w dostatecznie długim przedziale czasowym — po przeprowadzeniu dostatecznej liczby rund (por. rozdział 3.1) — to sędzia testujący gracza-maszynę odniesie sukces, a gracz-maszyna — porażkę.

Test Turinga umożliwia zatem sędziemu dokonanie trafnej identyfikacji gracza-maszyny. Okazuje się jednak, że pod pojęciem gracza-maszyny tak naprawdę kryje się dowolny system poznawczy korzystający w teście Turinga wyłącznie z metod algorytmicznych (w rozumieniu przyjętym w tych rozważaniach). Skoro tak, to może to być zarówno sztuczny system poznawczy, jak i naturalny system poznawczy, np. człowiek. W tym kontekście traci sens bardzo rozpowszechniony sposób mówienia o TT jako o teście odróżniającym ludzi od maszyn. Należałoby raczej mówić, że TT jest adekwatny jako narzędzie, które umożliwia odróżnianie systemów poznawczych (podkreślmy to jeszcze raz: bez rozróżniania na systemy sztuczne i naturalne) posługujących się wyłącznie metodami algorytmicznymi od tych, które korzystają z metod wykraczających poza algorytmy. Warto również zauważyć, że wspomniana adekwatność testu Turinga nie była — jak się wydaje — głównym celem, dla jakiego zaprojektowany został test. Celem tym było przecież zaprojektowanie testu, który pozwalałby na badanie obecności inteligencji w sztucznych systemach poznawczych (por. dwie tezy testu Turinga wg R. Frencha — strona 31). W kontekście Twierdzenia 2 rodzi się pytanie, czy TT jest dobrym narzędziem dla tak zaprojektowanego celu? (Oczywiście należy mieć na uwadze, że pytanie to ma sens w granicach ściśle zarysowanych założeń i ustaleń przyjętych powyżej). Okazuje się bowiem, że — przy spełnieniu pewnych warunków — gracz-maszyna zawsze może zostać rozpoznany jako maszyna właśnie, prze-

grywając tym samym grę w naśladownictwo. W teoretycznej perspektywie stawia to na równi maszyny, o których intuicyjnie orzeklibyśmy, że wykazują wiele inteligentnych zachowań oraz te, o których orzeklibyśmy, że wykazują minimalną liczbę takich zachowań. Jeśli jednak wyobrażamy sobie praktyczne przeprowadzanie testu Turinga, to wydaje się, że wspomniane różnice dotyczące graczy-maszyn mogłyby jednak zostać dostrzeżone (pewne wyobrażenia na ten temat daje nam przeprowadzany regularnie konkurs Loebnera).

Twierdzenie 2, rozważane w kontekście testu Turinga, ma pewne kłopotliwe konsekwencje. Po pierwsze, nie sposób *a priori* określić interwału czasowego, w którym sędzia odniesie sukces, trafnie identyfikując gracza-maszynę jako maszynę właśnie. Po drugie — i ważniejsze! — nie sposób określić momentu, w którym sędzia testujący gracza-człowieka powinien zakończyć test, zasadnie identyfikując gracza jako człowieka właśnie. Sędzia otrzymujący wyłącznie oczekiwane odpowiedzi może zawsze przypuszczać, że ma po prostu do czynienia z maszyną, której procedury umożliwiają generowanie odpowiedzi trafnych z uwagi na warunki stawianych przez niego zadań, jako że pytanie, na które trafna (z uwagi na warunki zadania) odpowiedź bezpośrednia nie może zostać wygenerowana przez maszynę nie zostało jeszcze zadane.

Twierdzenie 2 wydaje się mieć również pewne konsekwencje dla grupy argumentów, które określiłem jako argumenty z zupełnego systemu konwersacyjnego (por. rozdział 2.2). Przypomnijmy, że ten typ argumentacji sprowadza się do wykazania, że nawet jeśli maszyna zdałaby test Turinga, to i tak nie moglibyśmy powiedzieć o niej, że jest inteligentna, ponieważ TT dostarcza z gruntu błędnego kryterium posiadania inteligencji. Block, Lem i Searle opisują (hipotetyczne) sztuczne systemy poznawcze, które, opierając się wyłącznie na zaimplementowanych w nich algorytmach, mogą osiągnąć sukces w TT, rozumiany jako udzielanie sędziemu wyłącznie trafnych (z uwagi na warunki zadania ustalane przez sędziego) odpowiedzi bezpośrednich na zadawane tym systemom pytania. W propozycji Blocka podstawą działania odpowiednich algorytmów jest drzewo konwersacji (por. rysunek 2.1), kosmiczny gramofon Lema działa w oparciu o zasady określające, kiedy użyć jakiej płyty (por. rozdział 2.2.1), zaś w chińskim pokoju algorytmy określane są poprzez książkę przekładu (por. rozdział 2.2.2). Jedynie N. Block szkicuje bardziej szczegółowo metodę „pozyskiwania” odpowiednich algorytmów, pozostali wymienieni badacze po prostu przyjmują, że one istnieją (co jest krokiem dopuszczalnym, jako że mamy tu do czynienia z eksperymentami myślowymi). Teza o istnieniu algorytmu czy algorytmów, którego/których działanie umożliwi, dla każdego pytania, jakie może zadać sędzia, i dla każdych warunków zadania ustanowionych przez sędziego dla tego pytania, udzielanie wyłącznie trafnej — z uwagi na warunki zadania — odpowiedzi bezpośredniej na to pytanie, zdaje się pełnić kluczową rolę w dowolnym z argumentów z zupełnego systemu kowersacyjnego. Twierdzenie 2 wprowadza tu jednak pewne wątpliwości dotyczące możliwości istnienia takiego algorytmu. Oczywiście bezpośrednie zastosowanie rozważanego twierdzenia w tym kontekście nie jest możliwe. W omawianych eksperymentach myślowych nie możemy bowiem wprost mówić o prostym systemie pytań i odpowiedzi (tak jak w przypadku zrekonstruowanego w tej pracy testu Turinga). Przykłady podawane

przez N. Blocka, S. Lema czy J. Searle'a sugerują, że należałoby tutaj mówić raczej o odpowiednich reakcjach werbalnych na wypowiedzi pytającego. Co istotne, wydaje się jednak, że nawet w takiej sytuacji, hipotetyczne sztuczne systemy poznawcze rozpatrywane w argumentach z zupełnego systemu konwersacyjnego powinny — jak się wydaje — posiadać własności epistemiczno-pragmatyczne omówione w kontekście rozważań poświęconych konsekwencjom Twierdzenia 2 dla testu Turinga (por. s. 50).

Rozdział 3

Test Turinga — perspektywa sędziego

W niniejszym rozdziale spróbuję spojrzeć na test Turinga z perspektywy sędziego. Taka perspektywa jest, moim zdaniem, bardzo ważna dla analizy testu Turinga. Wskazują na to uwagi dotyczące roli sędziego w teście poczynione przez samego Turinga, a także trafne obserwacje N. Blocka dotyczące tego zagadnienia (por. rozdział 1). Niestety, tego typu rozważania nie są zbyt rozpowszechnione w literaturze przedmiotu. Warto wspomnieć tutaj o pracy „Undecidability in the Imitation Game" [Sato, Ikegami 2004]. Głównym celem autorów tego artykułu jest analiza roli i możliwości sędziego w teście Turinga. Przyjmują oni jednak — dość zaskakująco — że sędzia będzie modelowany przez maszynę (konkretnie przez maszynę Turinga). Sprawia to, że wynik otrzymany przez autorów tego artykułu jedynie pośrednio odnosi się do oryginalnej propozycji testu Turinga.

W tym rozdziale poruszone zostaną dwie kwestie związane z perspektywą sędziego: znaczenie doboru sędziego w teście Turinga oraz problem istnienia optymalnej strategii postępowania dla sędziego, dzięki której mógłby on dokonać trafnej identyfikacji gracza w TT.

W sprawie doboru sędziego wypowiada się już sam Turing w „Computing machinery...". Pisze, że powinien to być „przeciętny pytający" (*average interrogator*) [Turing 1950, s. 442]. Z kolei, w wywiadzie dla BBC z roku 1952 podkreśla, że nie powinna to być osoba, która „jest ekspertem w sprawie komputerów" [Newman et al. 1952, s. 4]. Alan Turing doskonale zdawał sobie sprawę z tego, że wiedza, którą posiada sędzia może rzutować na podejmowane przez owego sędziego decyzje. Dlatego właśnie wprowadza takie ograniczenie. Należy pamiętać, że jedno z założeń leżących u podstaw TT głosi, że rozmowa pomiędzy sędzią a uczestnikiem testu powinna przebiegać jak najbardziej naturalnie (tak, jak toczone są nasze codzienne rozmowy). Poruszaną kwestię podejmuje również N. Block w swoim artykule „The Mind as the Software of the Brain". Przypomnijmy tu — cytowany już w pierwszym rozdziale tej książki — odpowiedni fragment:

> Sędzia, który byłby wybitnym autorytetem w sprawie rzeczywiście inteligentnych maszyn, wiedziałby w jaki sposób odróżnić je od ludzi. Przykładowo, taki ekspert może wiedzieć, że obecne inteligentne maszyny radzą sobie z pewnymi problemami kłopotliwymi dla ludzi [Block 1995*b*, s. 379].

Dalej czytamy:

> Ludzi, którzy nie są zbyt obeznani z komputerami można zadziwiająco łatwo oszukać [...] [Block 1995*b*, s. 379].

Z przytoczonych fragmentów można wnioskować, że zarówno Turing, jak i Block zdawali sobie sprawę z tego, że dobór sędziego może wpłynąć na ostateczny wynik testu Turinga.

Problem doboru sędziego jawi się jako szczególnie istotny w kontekście konkursu Loebnera. Już sama formuła konkursu, w którym zwycięzca otrzymuje nagrodę pieniężną, sprawia, że poruszane tutaj zagadnienie wysuwa się na pierwszy plan. Co więcej, sami programiści, którzy biorą udział w konkursie Loebnera przyznają, że pewne stosowane przez nich rozwiązania opierają się właśnie na wiedzy o tym, kto będzie sędzią w danej edycji konkursu (por. [Mauldin 1994], [Garner 2009], [Humphrys 2009]).

Interesujący jest również fakt, że sama sytuacja konkursu nastawia sędziów bardziej podejrzliwie wobec uczestników. Można to zauważyć na przykładzie programu ChatterBot autorstwa M. Mauldina. Program zgłoszony został do konkursu Loebnera w 1993 roku, ale nie osiągnął w nim szczególnie wysokich wyników. Mauldin postanowił jednak sprawdzić, jak jego program poradzi sobie w grze TinyMUD (stworzonej przez J. Aspnesa). TinyMUD jest przygodową sieciową grą tekstową, w której uczestniczą gracze z całego świata. ChatterBot udawał jednego z takich graczy. Analiza zapisanych logów rozmów wskazuje na to, że program ten radził sobie znacznie lepiej niż w konkursie Loebnera. Warto tutaj podkreślić, że żaden z graczy-ludzi nie spodziewał się, że którakolwiek z napotkanych w wirtualnym świecie postaci nie będzie człowiekiem. M. Mauldin sugeruje, że być może taka forma testowania programów dialogowych (określa ją mianem *unsuspecting Turing test*) jest znacznie bardziej trafna niż ta wyznaczana przez ramy konkursu Loebnera.

O problemach, z jakimi borykają się organizatorzy konkursu Loebnera pisze szeroko Hugh Loebner (pomysłodawca konkursu) w artykule „How to Hold Turing Test Contest” [Loebner 2009]. Nieco dziwne wydaje się sugerowane przez Loebnera rozwiązanie poruszanej przez nas kwestii — poleca on, aby na sędziów wybierać dziennikarzy. Jego zdaniem bowiem, to oni najlepiej nadają się do tej roli ze względu na swoją „inteligencję i dociekliwość” (por. [Loebner 2009]). Pomysł ten zrealizowano w 1993 roku. Spotkał się on jednak z krytycznymi głosami ze strony programistów biorących udział w tej edycji konkursu (por. [Garner 2009] [Mauldin 1994]). R. Garner [2009] proponuje, aby sędziowie stanowili reprezentatywną próbkę całego społeczeństwa. Jego zdaniem zapewni to bardziej wiarygodne wyniki w konkursie Loebnera. Wspomniany artykuł R. Garnera jest jednak ciekawy przede wszystkim ze względu na swój główny temat. Autor opisuje w nim aplikację o nazwie Turing Hub. Jej głównym zadaniem jest pośredniczenie pomiędzy uczestnikami konkursu Loebnera a sędziami. Celem rozwoju tej aplikacji jest wypracowanie i implementacja standardowego interfejsu dla przeprowadzania konkursu Loebnera. Ów interfejs uniemożliwia stosowanie pewnych sztuczek programistycznych, które pozwoliłyby na oszukanie sędziego. Turing Hub ujednolica

format tekstu wpisywanego przez uczestników konkursu (zarówno maszyny, jak i ludzi), dzięki czemu sędziowie koncentrują się na treści otrzymywanych wypowiedzi, a nie na ich formie graficznej, sposobie, w jaki się pojawiają na monitorze itp.

W podobnym kierunku — choć już na znacznie wyższym poziomie ogólności — zmierza S. Watt w artykule „Can People Think? Or Machines? A Unified Protocol for Turing Test" [Watt 2009]. Autor proponuje w nim szkic uniwersalnego protokołu, wedle którego sędzia powinien przeprowadzać test Turinga. Przy konstrukcji tego protokołu Watt opierał się z jednej strony na badaniach bibliograficznych, z drugiej zaś na wynikach sondażu, który przeprowadził na potrzeby wspomnianej publikacji. Celem tych ustaleń było opracowanie skali cech, które mogą świadczyć o obecności inteligencji u danego podmiotu poznawczego (najwyżej w skali znalazły się autonomiczność, responsywność, zorientowanie na cel oraz komunikatywność — por. [Watt 2009, s. 309]). Niestety na podstawie samego tylko artykułu nie można zweryfikować metodologicznej wartości przedstawionych wyników (S. Watt nie podaje bowiem wielu niezbędnych do tego celu szczegółów zastosowanych procedur badawczych). Sam autor traktuje je jednak jako formę wstępnego szkicu i propozycję do dalszych rozważań. Protokół dla TT zawiera 14 pytań-wskazówek dla sędziego, na podstawie których może on formułować pytania do gracza. Są to m. in. następujące pytania (por. [Watt 2009, s. 315]):

— Czy uczestnik udziela odpowiedzi w czasie rzeczywistym?
— Czy uczestnik wykazuje się znajomością wiedzy potocznej (zdroworozsądkowej)?
— Czy uczestnik jest w stanie okazywać emocje?
— Czy uczestnik przypisuje innym posiadanie stanów mentalnych?

Dodatkowo protokół reguluje czas trwania TT, sposób komunikacji uczestnika z sędzią, a także pewne kwestie dotyczące samego sędziego (np. czy może on powoływać jako doradcę eksperta do spraw komputerów, aby zidentyfikował on strategie wykorzystywane przez uczestnika testu). Propozycja S. Watta — choć znajduje się na wczesnym etapie opracowywania — wydaje się ciekawym rozwiązaniem problemu doboru sędziego (a także krokiem w kierunku ustalenia dla niego strategii postępowania — por. np. rozdział 3.1). Jeżeli udałoby się opracować zestandaryzowany protokół przeprowadzania testu Turinga, to wpływ doboru sędziego na wynik TT zostałby zminimalizowany. Środek ciężkości tego problemu zostałby przeniesiony na metodę doboru pytań do takiego protokołu.

Obydwa opisane powyżej teksty zostały opublikowane w książce *Parsing the Turing Test* [Epstein et al. 2009], w części o bardzo znaczącym — w tym kontekście — tytule: „The New Methodological Debate". Rzeczywiście wydaje się, że wyznaczają one nową jakość w dyskusjach nad testem Turinga.

Zagadnienie strategii, jaką może stosować uczestnik TT, jest często poruszane w literaturze przedmiotu. Już sam Turing nakłada tutaj określone warunki opisane szczegółowo w rozdziale pierwszym niniejszej książki. Naturalne jest również to, że rozważania dotyczące tej kwestii pojawiają się częstokroć w tekstach autorów przygotowujących programy do udziału w konkursie

Loebnera. Zastanawiające jest to, że zagadnienie istnienia strategii dla sędziego jest niemalże nieobecne we wspomnianej literaturze. Pytanie, w jaki sposób sędzia mógłby sobie zapewnić, że dokona trafnej identyfikacji gracza w TT wydaje się — moim zdaniem — ze wszech miar ciekawe.

Próby odpowiedzi na nie spróbuję udzielić, korzystając z formalnego modelu TT sformułowanego w ramach inferencyjnej logiki pytań (*Inferential Erotetic Logic* — IEL) autorstwa A. Wiśniewskiego.

3.1. Próba formalnego ujęcia testu Turinga

Próby skonstruowania formalnego modelu testu Turinga lub choćby zastosowania pewnych narzędzi formalnych do jego zbadania nie są zbyt popularne w literaturze przedmiotu. W tym kontekście możemy wymienić artykuł [Sato, Ikegami 2004], w którym autorzy starają się przedstawić konsekwencje ograniczeń maszyn Turinga dla testu Turinga. Warto również wspomnieć tekst [Hernandez-Orallo 2000], w którym wykorzystuje się pewne osiągnięcia z dziedziny złożoności obliczeniowej dla zbadania testu Turinga jako testu posiadania inteligencji. Pełny formalny model testu Turinga — opierający się na teorii dowodów interakcyjnych — znajdziemy w [Bradford, Wollowski 1995], [Shieber 2007], a także w [Shieber 2006]. W porównaniu z ilością filozoficznej literatury na temat TT, sytuacja na polu formalizacji testu nie przedstawia się zbyt dobrze. Głównym powodem takiego stanu rzeczy jest — jak się wydaje — brak zgody co do rzeczywistych, oryginalnych założeń testu Turinga.

Omówię tutaj propozycję modelu testu Turinga bazującą na pragmatycznej interpretacji pewnych narzędzi dostarczanych przez inferencyjną logikę pytań. Wykorzystanie w tym miejscu modelu formalnego zapewnia, moim zdaniem, większą precyzję rozważań. Oczywiście kosztem jest tutaj konieczność poczynienia pewnych założeń początkowych wynikających z zastosowanych narzędzi formalnych. Wydaje się jednak, że pomimo tych ograniczeń udało się zachować intuicje związane z oryginalnym sformułowaniem testu Turinga.

Przy rekonstrukcji oryginalnych założeń dotyczących testu Turinga (por. rozdział 1) zauważyłem, że na TT możemy patrzeć jako na system pytań i odpowiedzi. W teście bierze udział dwóch uczestników: poddawany testowi (A) oraz sędzia (pytający — C). Można metaforycznie powiedzieć, że to na barkach sędziego spoczywa przebieg testu, ponieważ to on decyduje, jakie pytania zadać i wyłącznie na podstawie odpowiedzi udzielonych przez A musi zadecydować czy A jest człowiekiem, czy maszyną. Ponadto — jeśli przyjrzymy się przykładom przebiegu TT podawanym przez Turinga — sposób zadawania pytań, ich sformułowanie i dobór wydają się kluczowe w TT. Spostrzeżenia te skłaniają do próby zbadania testu Turinga rozpatrywanego z perspektywy sędziego oraz strategii, jakie może on przyjmować podczas przeprowadzania testu. W tym celu posłużę się inferencyjną logiką pytań autorstwa A. Wiśniewskiego. Główną motywacją wyboru właśnie tego aparatu formalnego jest

to, że inferencyjna logika pytań koncentruje się wokół zagadnienia rozumowań z wykorzystaniem pytań oraz warunków poprawności takich rozumowań. Jak zobaczymy w dalszej części niniejszego rozdziału, precyzyjne ujęcie tych zagadnień ma kapitalne znaczenie z perspektywy sędziego w teście Turinga. Pewne intuicje zaczerpnięte zostały również z matematycznej teorii gier oraz teorii dowodów interakcyjnych. Model przedstawiony poniżej nazwiemy dla uproszczenia TT_{IEL}.

3.1.1. TT jako gra

Przy formułowaniu TT_{IEL} wykorzystane zostaną następujące założenia dotyczące testu Turinga:

1. W grze uczestniczy dwóch graczy: C, czyli pytający, oraz gracz A, poddawany testowi.
2. A i C nie mogą się widzieć, słyszeć, pisać do siebie pismem odręcznym.
3. To C zadaje pytania, zaś A na nie odpowiada.
4. W pierwszej kolejności lepiej jest rozważać TT z pytaniami rozstrzygnięcia (por. [Turing 1950, s. 445] oraz rozdział 1).
5. Celem gracza A jest wprowadzenie w błąd gracza C, tak żeby dokonał on niepoprawnej identyfikacji (por. [Turing 1950, s. 434]). A zobowiązany jest również do postępowania zgodnie ze strategią jak najwierniejszego naśladownictwa odpowiedzi, jakich udzieliłby człowiek na jego miejscu.
6. Test ma charakter statystyczy i powinien być powtarzany kilkakrotnie (por. rozdział 1).

Tak ujęty TT traktowali będziemy jako rodzaj gry, w której ścierają się interesy dwóch graczy. Nawiązujemy tym samym do pojęcia gry, obecnego w matematycznej teorii gier. Przyjmuje się w niej, że o grze możemy mówić wszędzie tam, gdzie [Straffin 2001, s. 1]:

1. „Można wskazać co najmniej dwóch graczy. Graczem może być człowiek, ale także firma, państwo, czy nawet gatunek w znaczeniu biologicznym.
2. Każdy gracz ma do wyboru pewną liczbę możliwych *strategii*, określających sposób rozgrywania przez niego gry.
3. *Wynik* gry jest determinowany przez kombinację strategii wybranych przez poszczególnych graczy.
4. Każdemu możliwemu wynikowi gry odpowiada zestaw *wypłat* dla poszczególnych graczy, których wysokość można wyrazić liczbowo".

W dalszej części tego rozdziału — wzorując się na teorii dowodów interakcyjnych (por. [Goldwasser et al. 1985], [Papadimitriou 2002]) — graczy w TT_{IEL} będę nazywał, odpowiednio, „dowodzicielem" (*prover* — P) oraz „weryfikatorem" (*verifier* — V). Są oni odpowiednikami gracza A i C w oryginalnym sformułowaniu testu. Wykorzystanie innych oznaczeń motywowane jest tym, że lepiej oddają one intuicje związane z rolami graczy w TT niż tradycyjnie używane oznaczenia literowe A, C. Gracz P stara się przekonać

V co do prawdziwości sądu, że jest on człowiekiem a nie maszyną. Zadaniem V jest zweryfikowanie owego sądu. Możemy dookreślić, że TT modeluje sytuację czystego konfliktu, ponieważ nie ma tu mowy o kooperacji (współpracy) pomiędzy graczami. Każdy TT kończy się zwycięstwem jednego z graczy i przegraną drugiej strony. Sytuacja remisu nie jest tu możliwa, ponieważ TT kończy się każdorazowo werdyktem wydanym przez sędziego, w którym identyfikuje on gracza P albo jako maszynę albo jako człowieka. I tak:

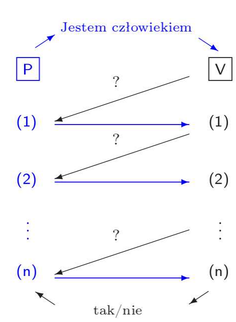
— P zwycięża, gdy V dokona nietrafnej identyfikacji.
— V zwycięża, gdy dokona trafnej identyfikacji.

Traktując TT jako pewnego rodzaju grę możemy przyjąć założenia dotyczące strategii graczy. Gracz P ma odpowiadać tak, jak odpowiadałby człowiek. Można powiedzieć, że P obowiązuje warunek „szczerości" (odpowiadania zgodnie z posiadaną przez siebie wiedzą). Przypomnijmy w tym miejscu cytowany już fragment „Computing Machinery...":

> Niektórzy mogą argumentować, że najlepszą strategią dla maszyny podczas „gry w naśladownictwo" mogłoby być coś innego niż naśladowanie zachowania człowieka. Być może tak jest, ale uważam, że jest mało prawdopodobne aby tego typu działania przyniosły jakiś znaczący efekt [...] Zakładam, że najlepszą strategią w tej sytuacji jest udzielanie odpowiedzi takich, jakie w naturalny sposób udzielone zostałyby przez człowieka [Turing 1950, s. 435].

Gracz V zadający pytania dąży do identyfikacji gracza P. Wydaje się, iż naturalne jest tutaj przyjęcie założenia, że dobór pytań zadawanych przez V nie będzie przypadkowy, ale podlegał będzie pewnej strategii. Jej wybór uzależniony jest od dwóch czynników: wiedzy początkowej gracza V (czyli zasobu wiedzy sędziego, zbioru jego przekonań na temat tego, co to znaczy być człowiekiem etc.) oraz odpowiedzi udzielanych przez P w trakcie testu.

Podsumowując, TT traktował będę jako grę składającą się ze skończonej liczby „rund", gdzie na każdą rundę składa się pytanie zadane przez V i odpowiedź nań udzielona przez P. Po ostatniej rundzie V dokonuje identyfikacji, czyli uznaje bądź odrzuca twierdzenie zgłaszane przez P na początku gry (warto przypomnieć, że zgodnie z założeniami TT to V decyduje, kiedy zakończyć grę). Ilustruje to rysunek 3.1.



Rysunek 3.1. TT jako gra. Gracz P stara się przekonać V co do prawdziwości sądu, że jest on człowiekiem. Sędzia V w kolejnych rudach gromadzi dane, które pozwolą mu uznać ten sąd za prawdziwy lub fałszywy (co kończy grę)

W prezentowanym podejściu koncentruję się głównie na perspektywie V w teście Turinga. Traktuję tym samym gracza P jako swego rodzaju „czarną skrzynkę” udzielającą odpowiedzi. Moim zdaniem doskonale oddaje to intuicje oryginalnej propozycji Turinga — nie jest istotne z jakich mechanizmów czy procedur korzysta P, to co się liczy, to wyłącznie odpowiedzi, których udziela.

Zanim przejdę do omówienia proponowanego tutaj modelu testu Turinga, dokonam krótkiego wprowadzenia do wykorzystywanego w nim aparatu formalnego.

3.1.2. Inferencyjna logika pytań i scenariusze erotetyczne

Na potrzeby niniejszej pracy zrezygnuję z bardzo szczegółowej, formalnej charakterystyki scenariuszy erotetycznych (w skrócie e-scenariuszy), ograniczając się do podania jedynie niezbędnych definicji. Czytelnika zainteresowanego większą ilością szczegółów odsyłam do prac twórcy e-scenariuszy — Andrzeja Wiśniewskiego ([Wiśniewski 2001], [Wiśniewski 2003], [Wiśniewski 2004]).

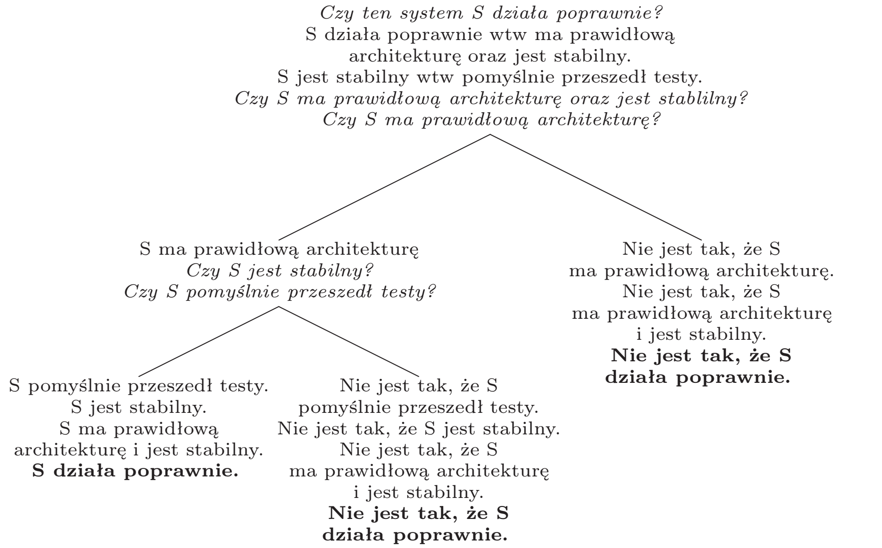
Najogólniej rzecz ujmując, idea e-scenariuszy stanowi realizację tzw. zasady dekompozycji (*Decomposition Principle — DP*). Zasadę tę — która wywodzi się z metody kartezjańskiej — możemy sformułować następująco:

> **DP:** *Problem początkowy powinien zostać rozłożony na prostsze problemy cząstkowe, w taki sposób, żeby rozwiązania owych problemów cząstkowych mogły złożyć się na rozwiązanie problemu początkowego* [Urbański, Wiśniewski 2006, s. 1].

Wyobraźmy sobie, na przykład, że zastanawiamy się nad tym, czy jakiś system działa poprawnie. Załóżmy też, że wiemy, że systemy działają poprawnie wtedy i tylko wtedy, gdy mają prawidłową architekturę i działają stabilnie. Jakie pytania powinniśmy zadać i w jakiej kolejności, aby uzyskać rozwiązanie naszego problemu tak szybko, jak to tylko możliwe? Rozwiązania dostarcza właśnie e-scenariusz. E-scenariusz możemy przedstawić w postaci drzewa, gdzie korzeniem jest pytanie początkowe, zaś liśćmi odpowiedzi bezpośrednie na owo pytanie. Scenariusz dla naszego przykładowego problemu widoczny jest na rysunku 3.2.

E-scenariusz z rysunku 3.2 można zapisać używając do tego sformalizowanego języka J (por. rysunek 3.3). Język J jest językiem klasycznego rachunku zdań (KRZ), którego słownik został rozszerzony o znaki: ?, {, }. Pojęcie formuły zdaniowej rozumiane jest tutaj tak, jak w KRZ. Znaki p, q, r, s, t, u, p_1, ... będą używane na oznaczenie zmiennych zdaniowych. Formuły zdaniowe języka J będą nazywane *formułami deklaratywnymi języka J* (jako metajęzykowych zmiennych dla formuł deklaratywnych użyte zostaną litery A, B, C, D, z ewentualnymi indeksami). Litery X, Y, Z (z ewentualnymi indeksami) będą wykorzystywane jako metajęzykowe zmienne dla oznaczenia zbiorów formuł deklaratywnych. *Pytaniem* języka J jest wyrażenie o postaci:

$$?\{A_1, A_2, ..., A_n\}$$



Rysunek 3.2. E-scenariusz dla przykładowego problemu (wyjaśnienia w tekście)

gdzie $n > 1$, zaś $A_1, A_2, ..., A_n$ są różnymi od siebie formułami deklaratywnymi. Każda z formuł $A_1, A_2, ..., A_n$ nazywana jest *odpowiedzią bezpośrednią* na pytanie o postaci $?\{A_1, A_2, ..., A_n\}$. Pytanie takie można czytać: „Czy jest tak, że A_1, lub czy jest tak, że A_2, ..., lub czy jest tak, że A_n?" Dla pewnych typów pytań przyjmujemy nieco inną konwencję notacyjną. Pytania typu: $?\{A, \neg A\}$ („Czy jest tak, że A?") zapisywali będziemy jako: $?A$. Pytania o schemacie $?\{A \wedge B, A \wedge \neg B, \neg A \wedge B, \neg A \wedge \neg B\}$ (tzw. pytania koniunkcyjne) zapisywali będziemy skrótowo jako: $? \pm |A, B|$. Pytanie takie możemy czytać jako „Czy jest tak, że A i czy jest tak, że B?" (por. [Wiśniewski 2003, s. 399])[1].

[1] Uogólniona definicja pytań koniunkcyjnych ma następującą postać (por. [Urbański 2001, s. 76]). Niech $A_1, ..., A_k$ $(k > 1)$ będą różnymi od siebie formułami deklaratywnymi. Niech $\alpha^{\mathrm{j}}(j = 1, ..., k)$ będzie ciągiem składającym się z 2^k wyrazów, którego n-ty element definiowany jest w sposób następujący:

$$\alpha_n^j = \begin{cases} A_j & \text{jeżeli } 1 \leq n \leq 2^{k-j} \\ \neg A_j & \text{jeżeli } 2^{k-j} < n \leq 2^{(k-j)+1} \\ \alpha_{n-m}^j & \text{jeżeli } 2^{(k-j)+1} < n \leq 2^k, \text{gdzie } m = 2^{(k-j)+1} \end{cases}$$

Niech $\beta^{\mathrm{i}}(1 \leq i \leq 2^k)$ będzie k-elementowym ciągiem zdefiniowanym następująco:

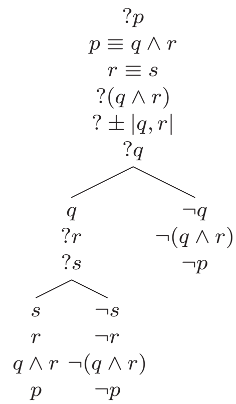
$$\beta^{\mathrm{i}}\langle \alpha_i^1, \alpha_i^2, ..., \alpha_i^k \rangle.$$

Pytanie koniunkcyjne z $A_1, ..., A_k$ jako czynnikami jest pytaniem o postaci: $?\{C_1, ..., C_t\}$, gdzie $t = 2^k$ i każde C_i $(i = 1, ..., t)$ ma postać

$$(\beta_1^i \wedge (\beta_2^i \wedge ...(\beta_{k-1}^i \wedge \beta_k^i)...).$$

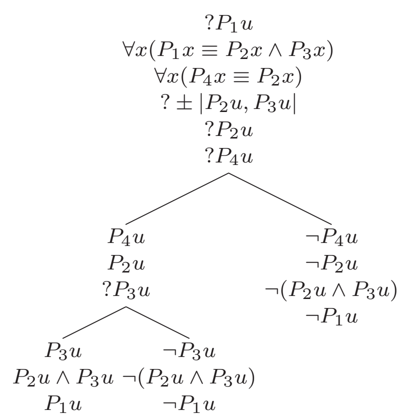
Pytanie takie będziemy skrótowo zapisywali jako $? \pm |A_1, ..., A_k|$.

Łatwo zauważyć, że e-scenariusz składa się ze ścieżek prowadzących od pytania początkowego, poprzez pytania cząstkowe (i odpowiedzi na nie), aż do odpowiedzi na pytanie początkowe.



Rysunek 3.3. E-scenariusz z rysunku 3.2 zapisany przy użyciu języka sformalizowanego J (opis języka w tekście)

Jeśli użyjemy bogatszego języka, wzrosną również możliwości modelowania pytań języka naturalnego. Jeśli np. wykorzystamy rachunek predykatów (dodając do niego wspomniane wcześniej stałe erotetyczne), możemy wyrazić m. in. pytanie o to, czy dany obiekt posiada cechę, którą jesteśmy zainteresowani (jest to oczywiście tylko przykład z całej grupy tzw. pytań pierwszego rodzaju — por. [Wiśniewski 1990, s. 30]). E-scenariusz, który rozpoczyna się właśnie takim pytaniem, przedstawia rysunek 3.4.



Rysunek 3.4. Przykład bardziej rozbudowanego e-scenariusza wyrażonego w bogatszym języku formalnym (opis w tekście). Pytanie początkowe tego e-scenariusza możemy zinterpretować jako: „Czy obiekt u posiada własność P_1?”

$$?\mathbf{S}(P_1x)$$
$$\forall x(P_1x \equiv P_2x)$$
$$\forall x(P_2x \to x = u_1 \vee x = u_2 \vee x = u_3)$$
$$P_2u_3 \to P_2u_1 \vee P_2u_2$$
$$P_2u_3$$
$$P_3u_1 \equiv P_2u_1$$
$$?\mathbf{S}(P_2x)$$
$$?\{P_2u_1, P_2u_2, P_2u_3\}$$
$$?\{P_2u_1, P_2u_2\}$$
$$?P_3u_1$$

$$P_3u_1 \quad \neg P_3u_1$$
$$P_2u_1 \quad P_2u_2$$
$$P_1u_1 \quad P_1u_2$$

Rysunek 3.5. Przykład bardziej rozbudowanego e-scenariusza wyrażonego w bogatszym języku formalnym (opis w tekście). Pytanie początkowe tego e-scenariusza możemy zinterpretować jako: „Który spośród x-ów posiada własność P_1?”

Dołączenie kolejnej stałej erotetycznej — **S** — pozwala na wyrażenie pytań o postaci:

$$?\mathbf{S}(Ax_{i_1},...x_{i_n}), \text{ gdzie } n \geqslant 1$$

.

W powyższej formule wyrażenie $Ax_{i_1},...x_{i_n}$ reprezentuje dowolną funkcję zdaniową, której wszystkimi zmiennymi wolnymi są $x_{i_1},...x_{i_n}$ (zakładamy, że zmienne $x_{i_1},...x_{i_n}$ są różne między sobą).

Odpowiedzi bezpośrednie na tego typu pytania przyjmują postać:

$$A(x_{i_1}/u_1, ..., x_{i_n}/u_n),$$

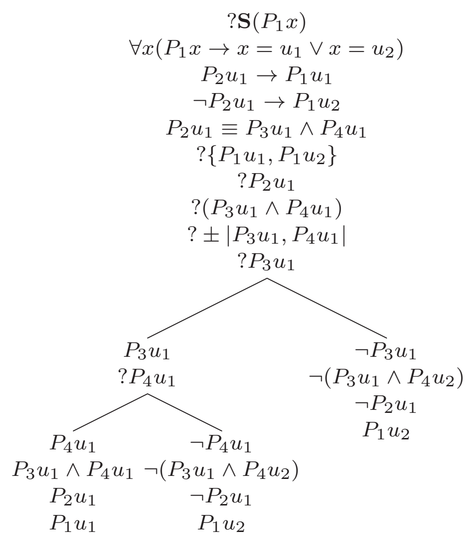
gdzie $u_1, ..., u_n$ są termami domkniętymi podstawionymi odpowiednio za zmienne wolne $x_{i_1}, ..., x_{i_n}$. Cechą charakterystyczną tego rodzaju pytań jest więc to, że są one pytaniami, w odpowiedzi na które należy wskazać przedmiot lub n-tkę przedmiotów spełniających daną funkcję zdaniową (por. [Wiśniewski 1990, s. 32]). I tak np. pytanie $?\mathbf{S}(P_1x)$ moglibyśmy odczytać jako: „Który spośród x-ów posiada własność P_1?” Scenariusze erotetyczne, w których pytania początkowe są właśnie pytaniami tego typu, przedstawiają rysunki 3.5 i 3.6.

Więcej szczegółów oraz konstrukcje pozwalające na wyrażanie innych pytań języka naturalnego zainteresowany czytelnik znajdzie np. w [Wiśniewski 1990] i [Wiśniewski 1995].

Kluczowe dla przedstawionych tutaj rozważań jest to, że dobór pytań pojawiających się na każdej ze ścieżek e-scenariusza nie jest przypadkowy. Są one powiązane ze sobą przez relację implikowania erotetycznego (por. [Wiśniewski 2003, s. 401]). W przypadku rozumowań erotetycznych relacja ta stanowi odpowiednik relacji wynikania dla formuł deklaratywnych (dostarczając kryterium poprawności rozumowań erotetycznych).

Definicja 1. *Pytanie* Q implikuje *pytanie* Q_1 ze względu na zbiór formuł deklaratywnych X *(symbolicznie:* $\mathbf{Im}(Q, X, Q_1)$*) wtw:*

1. *dla każdej odpowiedzi bezpośredniej A na pytanie Q: ze zbioru $X \cup \{A\}$ wynika wielownioskowo*[2] *zbiór odpowiedzi bezpośrednich na pytanie Q_1 oraz*
2. *dla każdej odpowiedzi bezpośredniej B na pytanie Q_1 istnieje niepusty podzbiór właściwy Y zbioru odpowiedzi bezpośrednich na pytanie Q, taki że Y wynika wielownioskowo ze zbioru $X \cup \{B\}$.*



Rysunek 3.6. Przykład bardziej rozbudowanego e-scenariusza wyrażonego w bogatszym języku formalnym (opis w tekście). Pytanie początkowe tego e-scenariusza możemy zinterpretować jako: „Który spośród x-ów posiada własność P_1?"

Warunek pierwszy powyższej definicji gwarantuje transmisję tzw. trafności pytań. Mówimy, że pytanie Q jest *trafne* wtedy i tylko wtedy, gdy co najmniej jedna odpowiedź bezpośrednia na to pytanie jest prawdziwa[3]. Intuicje zwią-

[2] W przypadku języka opartego na KRZ powiemy, że zbiór formuł deklaratywnych Y *wynika wielownioskowo* ze zbioru formuł deklaratywnych X wtw dla każdego wartościowania v, przy którym prawdziwe są wszystkie formuły należące do zbioru X, prawdziwa jest przynajmniej jedna formuła ze zbioru Y.

Gdy rozważamy język oparty na klasycznym rachunku predykatów, dla którego w klasie wszystkich interpretacji (części deklaratywnej) została wyróżniona niepusta podklasa (niekoniecznie właściwa) interpretacji standardowych, zbiór formuł deklaratywnych Y wynika wielownioskowo ze zbioru formuł deklaratywnych X wtw co najmniej jedna formuła ze zbioru Y jest prawdziwa przy każdej interpretacji standardowej (części deklaratywnej rozważanego języka), która jest modelem zbioru X.

Ogólna intuicja związana z wynikaniem wielownioskowym jest następująca: Y wynika wielownioskowo z X wtw co najmniej jedna formuła w Y musi być prawdziwa jeśli tylko wszystkie formuły w X są prawdziwe.

W sprawie wynikania wielownioskowego zob. [Shoesmith, Smiley 1978].

[3] Gdy rozważamy języki sformalizowane, pojęcie trafności ulega odpowiedniej relatywizacji. I tak, przykładowo, w przypadku języka, którego formułami deklaratywnymi są formuły KRZ, zrelatywizujemy trafność do wartościowania (zbioru formuł deklaratywnych). Definicja przyjmie postać następującą: Pytanie Q jest trafne przy wartościowaniu v (w skró-

zane z warunkiem drugim są następujące: każda z odpowiedzi bezpośrednich na pytanie Q_1 powinna w jakiś sposób zawężać klasę możliwości oferowanych początkowo przez cały zbiór odpowiedzi bezpośrednich na Q.

Dysponując definicją implikacji erotetycznej możemy teraz uszczegółowić intuicje związane z wyprowadzaniem odpowiedzi bezpośredniej na pewne pytanie. Takie wyprowadzenie nazywane jest derywacją erotetyczną (w skrócie e-derywacją). Pojęcie e-derywacji pozwala również na doprecyzowanie tego, jak zbudowane są ścieżki składające się na e-scenariusz.

Mówiąc dalej o formułach poprawnie zbudowanych, będę miał na myśli formuły deklaratywne oraz pytania rozważanego języka.

Definicja 2. *Skończony ciąg poprawnie zbudowanych formuł* $\mathbf{e} = \varphi_1, ..., \varphi_n$ *jest* derywacją erotetyczną odpowiedzi bezpośredniej A na pytanie Q z uwagi na zbiór formuł deklaratywnych X *wtedy i tylko wtedy, gdy* $\varphi_1 = Q$, $\varphi_n = A$ *i spełnione są następujące warunki:*

1. *dla każdego pytania* φ_k *ciągu* $\mathbf{e}$*, takiego że* $k > 1$*:*
   - *a.* $d_{\varphi_k} \neq dQ$*, oraz*
   - *b.* φ_{k+1} *jest albo pytaniem albo odpowiedzią bezpośrednią na* φ_k*;*
2. *dla każdej formuły deklaratywnej* φ_j *ciągu* $\mathbf{e}$
   - *a.* $\varphi_j \in X$*, lub*
   - *b.* φ_j *jest odpowiedzią bezpośrednią na* φ_{j-1}*, gdzie* $\varphi_{j-1} \neq Q$*, lub*
   - *c.* φ_j *wynika z pewnego zbioru formuł deklaratywnych takiego, że każdy element tego zbioru poprzedza* φ_j *w* $\mathbf{e}$*;*
3. *dla każdego pytania* φ_k *ciągu* $\mathbf{e}$*, takiego że* $\varphi_k \neq Q$*:* φ_k *jest implikowane przez pewne pytanie* φ_j*, które poprzedza* φ_k *w ciągu* $\mathbf{e}$ *z uwagi na zbiór pusty lub z uwagi na zbiór formuł deklaratywnych taki, że każdy element tego zbioru poprzedza* φ_k *w ciągu* $\mathbf{e}$*.*

Intuicyjnie możemy patrzeć na e-derywację jako na ciąg kroków prowadzących od pytania do odpowiedzi na nie. Przy czym każdy kolejny krok musi być „legalny". Pierwszy warunek definicji zapewnia, że pytania, które będą pojawiały się w e-derywacji po pytaniu początkowym nie będą z nim tożsame (innymi słowy, pytania te nie mogą mieć dokładnie takich samych zbiorów odpowiedzi bezpośrednich). Co więcej, formuła pojawiająca się w e-derywacji po pytaniu może być albo odpowiedzią bezpośrednią na to pytanie, albo kolejnym pytaniem. Warunek drugi definicji określa, kiedy formuła deklaratywna może pojawić się w e-derywacji. Są trzy takie przypadki: kiedy formuła deklaratywna należy do zbioru przesłanek początkowych e-derywacji (należy do zbioru formuł deklaratywnych, z uwagi na który konstruuje się e-derywację), gdy stanowi odpowiedź bezpośrednią na pytanie występujące w e-derywacji po pytaniu początkowym, lub kiedy wynika z pewnych formuł deklaratywnych występujących wcześniej w e-derywacji. Ostatni warunek mówi, że każde

cie: jest v-trafne) wtedy i tylko wtedy, gdy przynajmniej jedna odpowiedź bezpośrednia na Q jest prawdziwa przy wartościowaniu v (por. [Wiśniewski 2003, s. 400]).

pytanie pojawiające się po pytaniu początkowym musi być implikowane erotetycznie przez poprzedzający go element (lub elementy) e-derywacji (por. [Wiśniewski 2003, s. 403]).

Przypatrzmy się prostemu przykładowi e-derywacji dla pytania $?\{p,q,r\}$ z uwagi na zbiór formuł deklaratywnych $X = (s \vee t,\ s \rightarrow p,\ t \rightarrow q \vee p)$. Poszczególne elementy e-derywacji oddzielone są od siebie średnikami:

$$?\{p,q,r\}; s \vee t; s \rightarrow p; t \rightarrow q \vee r; ?\{s,t\}; s; p.$$

Wprowadzenie pytania $?\{s,t\}$ jest możliwe dzięki temu, że zachodzi

$$\mathbf{Im}(?\{A,B,C\}, D \vee E, D \rightarrow A, E \rightarrow B \vee C, ?\{D,E\}).$$

Z uwagi na dalsze rozważania poświęcone e-scenariuszom, ważnym pojęciem związanym z e-derywacją jest pojęcie zapytania (*query*). Zapytanie możemy zdefiniować następująco:

Definicja 3. *Element φ_k (gdzie $1 < k < n$) e-derywacji $\mathbf{e} = \varphi_1, ..., \varphi_n$ jest* zapytaniem (*query*) e-derywacji **e***, jeżeli φ_k jest pytaniem oraz φ_{k+1} jest odpowiedzią bezpośrednią na φ_k.*

I tak w powyższym przykładzie zapytaniem jest pytanie $?\{s,t\}$.

Dysponując definicjami implikacji erotetycznej oraz e-derywacji możemy przejść do podania definicji scenariusza erotetycznego, na który możemy spojrzeć jako na rodzinę e-derywacji.

Definicja 4. *Skończona rodzina ciągów Φ poprawnie zbudowanych formuł jest* scenariuszem erotetycznym dla pytania Q z uwagi na zbiór formuł deklaratywnych X *wtedy i tylko wtedy, gdy każdy element Φ jest e-derywacją odpowiedzi bezpośredniej na Q z uwagi na zbiór formuł deklaratywnych X i spełnione są poniższe warunki:*

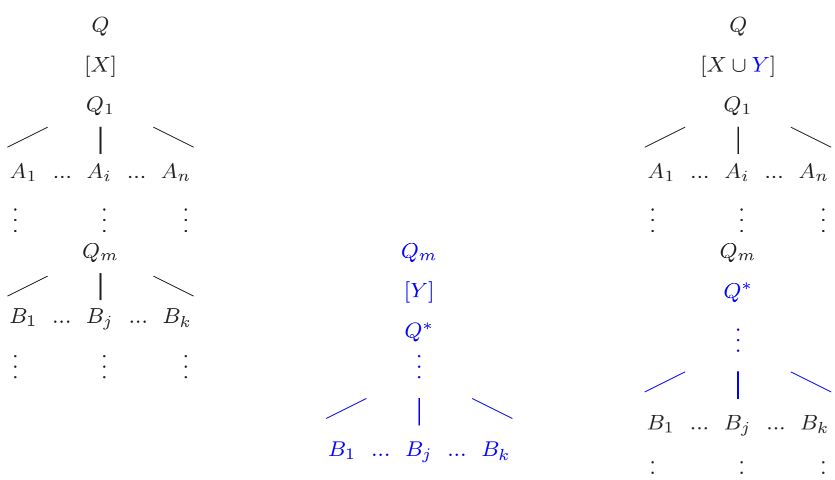
1. *$dQ \cap X = \emptyset$;*
2. *Φ zawiera przynajmniej dwa elementy;*
3. *dla każdego elementu $\mathbf{e} = \varphi_1, \varphi_2, ..., \varphi_n$ rodziny ciągów Φ, dla każdego k, takiego, że $1 \leq k < n$:*
   - a. *jeżeli φ_k jest pytaniem oraz φ_{k+1} jest odpowiedzią bezpośrednią na φ_k, to dla każdej odpowiedzi bezpośredniej B na φ_k rodzina ciągów Φ zawiera pewną e-derywację $\mathbf{e}' = \psi_1, \psi_2, ..., \psi_m$ taką, że $\psi_j = \varphi_j$ dla $j = 1, ..., k$ oraz $\psi_{k+1} = B$;*
   - b. *jeżeli φ_k jest formułą deklaratywną lub φ_k jest pytaniem oraz φ_{k+1} nie jest odpowiedzią bezpośrednią na φ_k, to dla każdej e-derywacji $\mathbf{e}' = \psi_1, \psi_2, ..., \psi_m$ w Φ takiej, że $\psi_j = \varphi_j$ dla $j = 1, ..., k$ mamy $\psi_{k+1} = \varphi_{k+1}$.*

E-scenariusze posiadają pewne ciekawe własności. Dwie z nich są szczególnie interesujące z perspektywy naszej pracy: pierwszą opisuje tzw. *twierdzenie o złotej ścieżce*; drugą jest możliwość modyfikacji e-scenariusza dla danego pytania początkowego. W przypadku języków, których formułami deklaratywnymi są formuły KRZ *twierdzenie o złotej ścieżce* ma następującą postać:

Twierdzenie 3. Twierdzenie o złotej ścieżce*: Niech Φ będzie e-scenariuszem dla pytania Q z uwagi na zbiór formuł deklaratywnych X. Niech v będzie takim wartościowaniem, że Q jest v-trafne i wszystkie formuły deklaratywne w X są prawdziwe przy wartościowaniu v. Wtedy scenariusz Φ zawiera przynajmniej jedną ścieżkę* **e** *taką że:*

(a) każda formuła deklaratywna w **e** *jest prawdziwa przy wartościowaniu v; oraz*
(b) każde pytanie w **e** *jest v-trafne; oraz*
(c) **e** *prowadzi do bezpośredniej odpowiedzi na Q, która jest prawdziwa przy wartościowaniu v* [Wiśniewski 2003, s. 411].

Jeżeli początkowe pytanie e-scenariusza jest v-trafne oraz wszystkie przesłanki początkowe są prawdziwe (przy wartościowaniu v), wtedy przynajmniej jedna ze ścieżek tego e-scenariusza prowadzi do prawdziwej (przy wartościowaniu v) odpowiedzi bezpośredniej na pytanie początkowe. Ścieżka ta zawiera jedynie trafne pytania cząstkowe i prawdziwe zdania (pośród nich prawdziwe odpowiedzi na pytania cząstkowe). Można więc powiedzieć, że e-scenariusz nie tylko przedstawia plan poszukiwania odpowiedzi na pytanie początkowe, ale przedstawia „bezpieczny plan" poszukiwania tej odpowiedzi, który dodatkowo jest skończony, czyli poszukiwanie odpowiedzi kończy się w skończonej liczbie kroków (por. [Wiśniewski 2004, s. 151]).



Rysunek 3.7. Schemat operacji wklejania jednego e-scenariusza do drugiego (na podstawie [Wiśniewski 2008])

Jak zauważyłem powyżej, e-scenariusz przedstawia plan poszukiwania odpowiedzi na pytanie początkowe. Z perspektywy tych rozważań szczególnie istotne jest to, że ów plan może zostać poddany modyfikacjom poprzez wykonanie prostych operacji na e-scenariuszach. Dzięki temu możliwe jest dopasowanie takiego planu do zaistniałych potrzeb. Co istotne, po dokonaniu

wspomnianych operacji e-scenariusze zachowują wszystkie interesujące nas własności. Przyjrzyjmy się bliżej jednej z takich operacji — wklejaniu e-scenariuszy (*systematic embedding*). Intuicyjnie mówiąc, możliwe jest wklejenie jednego e-scenariusza do drugiego, tak aby w efekcie otrzymać nowy e-scenariusz. Jeżeli na przykład mamy e-scenariusz Φ dla pytania Q zbudowany ze względu na zbiór przesłanek X oraz zapytanie (*query*) Q^* występujące na jednej ze ścieżek Φ, a także e-scenariusz Ψ dla pytania Q^* zbudowany w oparciu o zbiór przesłanek Y, to możemy wkleić Ψ do Φ (oczywiście jest to możliwe, gdy spełnione są odpowiednie warunki — zob. [Wiśniewski 2003, s. 413–414]). W wyniku tej operacji otrzymamy nowy e-scenariusz w oparciu o zbiór przesłanek będący sumą zbioru przesłanek dla scenariusza Φ i zbioru przesłanek dla scenariusza Ψ (por. [Wiśniewski 2004, s. 14]). Co istotne, nowy e-scenariusz przedstawia zmodyfikowany plan poszukiwania odpowiedzi dla pytania Q. Schematycznie tę operację przedstawia rysunek 3.7 (por. [Wiśniewski 2008]). Po lewej stronie mamy wyjściowy e-scenariusz dla pytania Q z uwagi na zbiór formuł deklaratywnych X. Na jednej ze ścieżek znajduje się zapytanie Q_m. Dysponujemy również e-scenariuszem dla pytania Q_m z uwagi na zbiór formuł deklaratywnych Y (środkowy e-scenariusz). Możemy wkleić e-scenariusz dla pytania Q_m do wyjściowego e-scenariusza. W efekcie otrzymamy nowy e-scenariusz (na rysunku znajdujący się po prawej stronie) dla pytania Q z uwagi na sumę zbiorów formuł deklaratywnych X i Y.

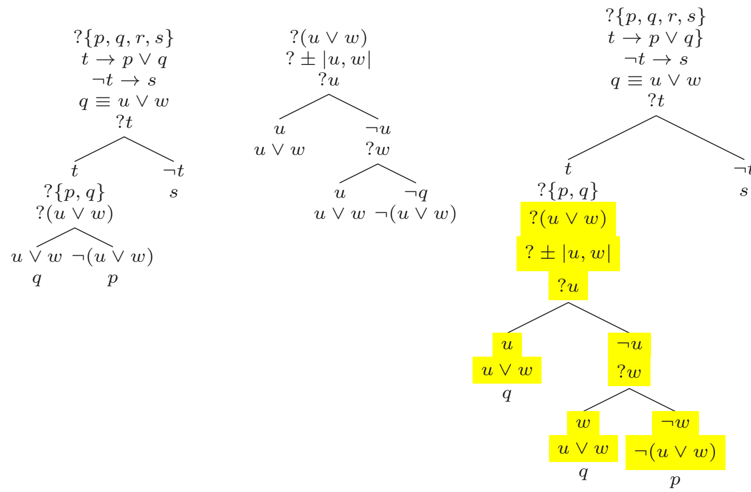
Przyjrzyjmy się teraz działaniu operacji wklejania na prostym przykładzie (por. [Wiśniewski 2004, s. 16]). Załóżmy, że dysponujemy e-scenariuszem dla pytania $?\{p, q, r, s\}$ — por. rysunek 3.8. Na jednej z jego ścieżek znajduje się zapytanie $?(u \vee w)$. Dysponujemy również drugim e-scenariuszem właśnie dla tego pytania (środkowy e-scenariusz na rysunku 3.8). Możemy zatem wkleić drugi e-scenariusz do pierwszego w miejsce zapytania $?(u \vee w)$. Otrzymujemy w rezultacie e-scenariusz przedstawiający zmodyfikowany plan poszukiwania odpowiedzi dla pytania $?\{p, q, r, s\}$.

Dokładne formalne ujęcie procedury wklejania zainteresowany czytelnik znajdzie w [Wiśniewski 2003, s. 412–413], a także w [Wiśniewski 2004, s. 154–156] (w postaci reguł diagramatycznych).

Procedura modyfikowania planów poszukiwań odpowiedzi przedstawianych przez e-scenariusze posiada bardzo intuicyjne motywacje. Dzięki operacji wklejania możliwe jest modyfikowanie pierwotnego planu poszukiwania odpowiedzi przy uwzględnieniu osiągniętych wcześniej wyników poszukiwań, nowych przesłanek itp. Ta właściwość e-scenariuszy sprawia, że wydają się one odpowiednie do modelowania strategii gracza V w teście Turinga.

3.1.3. Scenariusze erotetyczne a perspektywa sędziego w TT

Przypomnijmy, że w TT_{IEL} traktuję test Turinga jako grę składającą się ze skończonej liczby „rund". Na każdą z takich rund składa się pytanie sędziego V oraz odpowiedź udzielona na nie przez P. W tej grze V dąży do



Rysunek 3.8. Przykład operacji wklejania jednego e-scenariusza do drugiego

zweryfikowania prawdziwości twierdzenia zgłaszanego przez P mówiącego, że jest on człowiekiem. V dokonuje tej weryfikacji na podstawie odpowiedzi udzielanych przez P (por. rysunek 3.1). Założyliśmy na potrzeby TT_{IEL}, że pytania formułowane przez V nie będą przypadkowe. Naturalne wydaje się bowiem, że V przyjmie pewną strategię odpytywania gracza P. Przyjmijmy, że w tym celu V wykorzysta erotetyczne scenariusze poszukiwań.

Aby wykorzystać e-scenariusze do opisu TT, czy też dokładniej rzecz ujmując — do opisu TT z perspektywy sędziego, odróżnimy pytania *stawiane* sobie przez sędziego (V) od pytań *zadawanych* testowanemu (P) przez sędziego. Pytania, które stawia sobie sędzia, nie muszą być wprost komunikowane graczowi P (łatwo możemy sobie wyobrazić, że dość naiwne byłoby np. zadanie pytania: „Czy jesteś inteligentny?”). Naturalne wydaje się przyjęcie, że pytania, które stawia sobie sędzia w jakiś sposób będą wyznaczały pytania zadawane graczowi P. Aby ściślej wyrazić tę intuicję, odwołamy się tu do pragmatycznej interpretacji erotetycznych scenariuszy poszukiwań.

Zakładamy, że sędzia będzie wykorzystywał e-scenariusz jako strategię prowadzenia testu. Pod pojęciem strategii rozumiemy tutaj plan gry, który wyznacza, jakie pytania i w jakiej kolejności powinien zadać sędzia graczowi P (por. np. [Lewin et al. 1967], [Kelly 2003]).

Można zadać pytanie, dlaczego V miałby przyjąć jako strategię scenariusz erotetyczny. Oczywiście nie musi on tego robić, ale użycie e-scenariusza będzie dla niego korzystne. Wyżej przyjąłem, że V nie będzie raczej formułował pytań w sposób przypadkowy i że jego kolejne pytania będą zależały w jakiś sposób od odpowiedzi udzielanych przez P oraz od wiedzy i przekonań posiadanych przez gracza V. Warunki te spełnia e-scenariusz. Dostarcza on V informacji mówiących, jakie pytanie powinien postawić i kiedy (czyli po jakiej odpowiedzi udzielonej przez P). Ponadto dobór pytań w oparciu o e-scenariusz gwarantuje, że będą one zadawane zawsze w związku z py-

taniem początkowym, uniemożliwiając „zbaczanie z tematu". Kolejną cechą e-scenariuszy (w tym kontekście najważniejszą) jest to, że *gwarantują* one, iż każde postawione pytanie przybliża nas do uzyskania odpowiedzi na pytanie początkowe. Grając na podstawie e-scenariusza wiemy z całą pewnością, że odpowiedzi uzyskane na pytania cząstkowe złożą się ostatecznie na odpowiedź na pytanie początkowe.

Dodatkowym argumentem na rzecz wykorzystania e-scenariuszy jako strategii gry w TT jest własność opisywana przez twierdzenie o *złotej ścieżce*. W tym kontekście możemy powiedzieć, że dla strategii gry w TT określonej przez e-scenariusz istnieje przynajmniej jedna taka rozgrywka, która kończy się rozwiązaniem problemu początkowego i dodatkowo owo rozwiązanie jest poprawne (z uwagi na przyjęte założenia początkowe). Co więcej, możliwe jest w miarę proste i intuicyjne modyfikowanie pierwotnego planu poszukiwania przedstawianego przez e-scenariusz, chociażby przy użyciu procedury wklejania (zobacz s. 68).

Oczywiście należy pamiętać również o tym, że zawsze skorzystanie z narzędzia formalnego prowadzi do konieczności nałożenia pewnych ograniczeń i w konsekwencji do pewnych uproszczeń badanego zjawiska. W przypadku TT_{IEL} jesteśmy ograniczeni definicją implikacji erotetycznej. W zamian uzyskujemy jednak możliwość ścisłego badania TT widzianego z perspektywy sędziego (oczywiście na pewnym stopniu ogólności).

Zakładając, że sędzia będzie korzystał z e-scenariusza jako strategii w TT, możemy przyjąć, że odpowiednie przekonania sędziego znajdą swój wyraz w przesłankach rozważanego e-scenariusza. Będą to oczywiście przekonania sędziego odnośnie do kryteriów, które musi spełnić testowany gracz P, aby być uznanym za człowieka. Swoje przekonania sędzia może formułować w dwojaki sposób; albo będą one wyrażały warunki wystarczające „bycia człowiekiem", takie że niespełnienie żadnego z nich przekona sędziego, że poddawany testowi nie jest człowiekiem, albo też warunki konieczne, które łącznie — w przekonaniu sędziego — składają się na warunek wystarczający.

W pierwszym przypadku przesłanki będą formułowane wedle następującego schematu:

„Jeżeli gracz P spełnia kryterium X, to gracz P jest człowiekiem"

Możemy ten schemat zapisać jako $B \to A$, gdzie A skraca „gracz P jest człowiekiem", zaś B reprezentuje frazę „gracz P spełnia kryterium X". W takiej sytuacji przesłanki odzwierciedlające przekonania sędziego moglibyśmy wyrazić za pomocą zestawu formuł o schematach:

$B_1 \to A$
$B_2 \to A$
...
$B_n \to A$
$\neg B_1 \land \neg B_2 \land ... \land \neg B_n \to \neg A$

gdzie A jest różne od każdego B_i $(1 \leq i \leq n)$.

Sędzia dysponuje w tym przypadku warunkami wystarczającymi do „bycia człowiekiem", takimi że gdy żaden z nich nie jest spełniony, jest to w jego opinii warunkiem wystarczającym do tego, żeby uznać, że nie ma on do czynienia z człowiekiem.

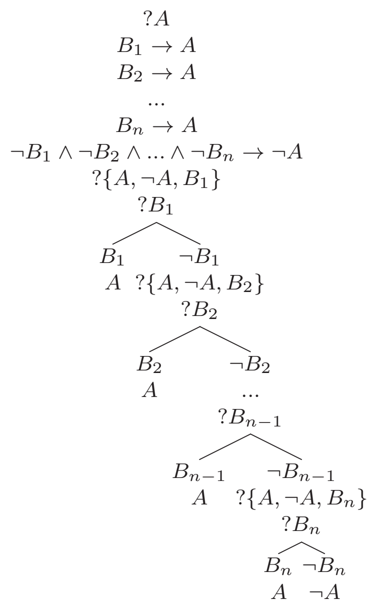
Z uwagi na to, że zachodzą następujące zależności:

(1) $\mathbf{Im}(?A, B_i \to A, ?\{A, \neg A, B_i\})$
(2) $\mathbf{Im}(?\{A, \neg A, B_i\}, ?B_i)$

możemy przejść od pytania $?A$ do pytania o postaci $?B_i$ ze względu na odpowiednią przesłankę o postaci $B_i \to A$.

Erotetyczny scenariusz poszukiwań zbudowany w oparciu o przesłanki tego typu podpadałby pod schemat:

(I)



Na przedstawionym powyżej schemacie e-scenariusza widzimy, że procedura postępowania sędziego jest następująca: najpierw stawia on sobie pytanie, czy gracz P jest człowiekiem, a następnie stawia sobie kolejno pytania o to, czy P spełnia kolejne kryteria „bycia człowiekiem"[4]. W sytuacji gdy P nie spełni żadnego z warunków, sędzia zyskuje pewność, że odpowiedź na główne pytanie realizowanego przez niego e-scenariusza jest negatywna.

W drugim z rozważanych wariantów, przesłanki formułowane przez sędziego podpadałyby pod następujący schemat:

„Jeżeli gracz P jest człowiekiem, to gracz P spełnia warunek X."

[4] Zauważmy, że pytania o postaci $?\{A, \neg A, B_i\}$ nie są zapytaniami rozważanego e-scenariusza, lecz pełnią jedynie rolę niezbędnych przesłanek, które z jednej strony są (erotetycznie) implikowane, z drugiej zaś implikują (erotetycznie) odpowiednie zapytania.

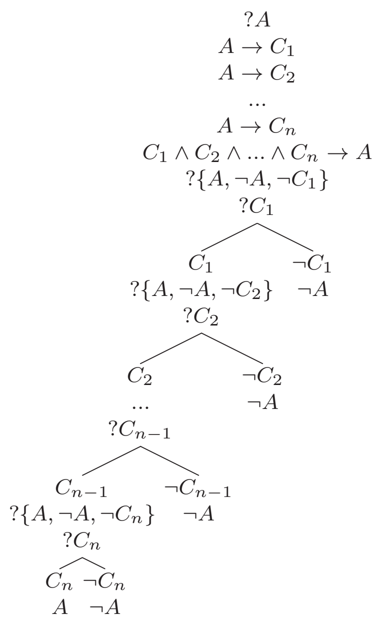
Analogicznie do poprzedniego rozwiązania (przy czym tutaj C reprezentuje frazę „gracz P spełnia kryterium X”), możemy teraz przedstawić przekonania sędziego jako zestaw formuł o schematach:

$A \to C_1$
$A \to C_2$
...
$A \to C_n$
$C_1 \wedge C_2 \wedge ... \wedge C_n \to A$

gdzie A jest różne od każdego C_i $(1 \leq i \leq n)$.

Sędzia dysponuje więc warunkami koniecznymi do „bycia człowiekiem”, których *łączne* spełnienie jest w jego opinii warunkiem wystarczającym do bycia człowiekiem. Scenariusz, wedle którego sędzia mógłby prowadzić test, podpadałby pod następujących schemat:

(II)



Zachodzą bowiem:

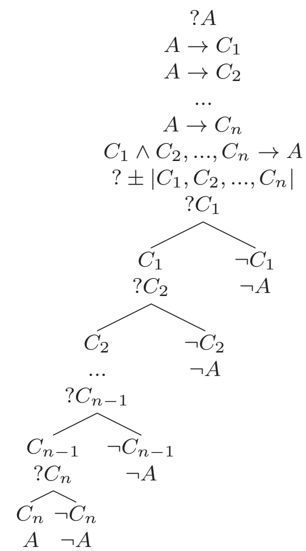
(3) $\mathbf{Im}(?A, A \to C_i, ?\{A, \neg A, \neg C\})$
(4) $\mathbf{Im}(?\{A, \neg A, \neg C_i\}, ?C_i)$.

Zachodzą również poniższe zależności:

(5) $\mathbf{Im}(?A, A \to C_1, A \to C_2, ..., A \to C_n, C_1 \wedge C_2 \wedge ... \wedge C_n \to A, ? \pm |C_1, C_2, ..., C_n|)$
(6) $\mathbf{Im}(? \pm |C_1, C_2, ..., C_n|, ?C_i)$, gdzie $1 \leq i \leq n$.

Przypomnijmy, że zapis $? \pm |C_1, C_2, ..., C_n|$ oznacza pytanie koniunkcyjne (zob. s. 62). Rozważany e-scenariusz mógłby więc także podpadać pod schemat następujący:

(III)



W przypadku obu powyższych schematów procedura postępowania sędziego w TT polega na sprawdzeniu kolejnych warunków sformułowanych przez sędziego. Gdy gracz P spełni wszystkie z nich, sędzia zyskuje pewność, że ma do czynienia z człowiekiem.

Warto nadmienić, że prawdopodobnie w rzeczywistym teście Turinga najkorzystniejsze byłoby zastosowanie strategii będącej połączeniem obu przedstawionych powyżej rozwiązań. Praktyczne zastosowanie tych rozwiązań wymagałoby również niewątpliwie wykorzystania pewnych elementów rozumowania statystycznego. Takie dodatkowe reguły statystyczne mogłyby np. określać dopuszczalne proporcje odpowiedzi uznanych i nieuznanych przez sędziego. Strategia opracowywania takich reguł mogłaby być na przykład wzorowana na rozwiązaniu R. Frencha zaproponowanym w [French 1996], a opartym na tzw. *Human Subcognitive Profile* (zob. rozdział 2). Formułowanie takich reguł i cyzelowanie strategii sędziego w kierunku rzeczywistych, praktycznych zastosowań wykracza jednak poza poziom ogólności rozważań niniejszej pracy.

Kolejne zapytania przedstawionych e-scenariuszy należy — jak się wydaje — traktować jako pytania stawiane sobie przez sędziego. Jak już zauważyliśmy wcześniej, zadawanie tych pytań graczowi P wydaje się zajęciem dość jałowym — bowiem jeśli otrzymamy na nie odpowiedź, znaczy to tylko tyle, że testowany sztuczny system potrafi podać deklaracje, na podstawie których sędzia (w oparciu o przyjęte kryteria) wyciągnie odpowiedni wniosek. Przykładowo, zadanie graczowi P pytania: „Czy potrafisz odpowiedzieć na pytanie subkognitywne?” (por. rozdział 2) i uzyskanie od niego odpowiedzi „tak” nie przyniesie sędziemu wiedzy o P rzeczywiście użytecznej z perspektywy testu Turinga. Z tego powodu uznajemy, że pytania zadawane sobie przez sędziego

— i w tym przypadku reprezentowane przez zapytania e-scenariuszy — jedynie *wyznaczają* pytania, które sędzia zada testowanemu graczowi P.

Aby sprecyzować owo wyznaczanie pytań przez zapytania e-scenariusza, dokonam pewnej operacjonalizacji. W tym celu zakładam, że sędzia przyjmuje konkretne przesłanki o schemacie:

(*) jeśli a jest człowiekiem oraz formułuję warunek w_i (będący warunkiem zadania) i następnie zadaję temu a odpowiednie pytanie Q_i, to a udziela odpowiedzi o_i na pytanie Q_i.

W powyższym schemacie przesłanki przyjmowanej przez sędziego, o_i reprezentuje odpowiedź na pytanie Q_i taką, że w opinii sędziego właśnie tej odpowiedzi „udzieliłby człowiek” z uwagi na warunek w_i odpowiedniego zadania.

Schemat ten będę dalej zapisywać skrótowo:

(**) $\mathbf{C}(a) \wedge \mathbf{F}(w_i, a, Q_i) \rightarrow \mathbf{U}(a, o_i, Q_i)$,

gdzie $\mathbf{C}(a)$ oznacza „a jest człowiekiem”, $\mathbf{F}(w_i, a, Q_i)$ — „formułuję warunek w_i (będący warunkiem zadania) i następnie zadaję a pytanie Q_i”, zaś $\mathbf{U}(a, o_i, Q_i)$ — „a udziela odpowiedzi o_i na pytanie Q_i”.

Zakładając, że takich przesłanek jest n (gdzie $n > 1$), strategia, którą będzie się teraz posługiwał sędzia, może się wyrażać e-scenariuszem podpadającym pod zaprezentowany poniżej schemat (A)[5].

Dzięki takiemu ujęciu możliwe jest wyraźne odróżnienie pytań zadawanych sobie przez sędziego od tych, które postawi on testowanemu graczowi P. Pytania, które zadaje sobie sędzia, to pytania $?\mathbf{U}(a, o_1, Q_1), ..., ?\mathbf{U}(a, o_n, Q_n)$, zaś pytania, które postawi on graczowi P to odpowiednie pytania $Q_1, ..., Q_n$.

Analogicznej operacjonalizacji można poddać e-scenariusz o schemacie (I) (zob. s. 72) z tym, że przesłanki przyjmowane przez sędziego podpadałyby wtedy pod schemat: $\mathbf{F}(w_1, a, Q_i) \wedge \mathbf{U}(a, o_i, Q_i) \rightarrow C(a)$.

Z uwagi na twierdzenie o złotej ścieżce możemy stwierdzić, że sędzia, realizując zaproponowany e-scenariusz, dokona trafnej identyfikacji gracza P. Oczywiście należy pamiętać, że stanie się tak pod warunkiem, że przesłanki deklaratywne tego e-scenariusza będą prawdziwe, co jest bardzo silnym założeniem.

Na tym etapie rozważań widać wyraźnie, że przebieg testu Turinga zależy w ogromnym stopniu od zasobu *wiedzy* i od *przekonań* sędziego. Zagadnienie to jest bardzo istotne dla TT (por. np. [Turing 1950, s. 442], [Newman et al. 1952, s. 4], [Block 1995*b*, 378-379]). Erotetyczny scenariusz poszukiwań gwarantuje jedynie to, że sędzia otrzyma odpowiedź na pytanie główne realizowanego e-scenariusza. Nie gwarantuje prawdziwości uzyskanej odpowiedzi, rozumianej jako trafna identyfikacja gracza P. Trafność tej identyfikacji zależy w oczywisty sposób od zestawu przesłanek, na podstawie których V buduje e-scenariusz dla testu Turinga.

---

[5] W schemacie korzystamy z tego, że zachodzą relacje implikacji (6) oraz

(7) $\mathbf{Im}(?A, A \wedge D_1 \rightarrow C_1, A \wedge D_2 \rightarrow C_2, ..., A \wedge D_n \rightarrow C_n, D_1 \wedge D_2 \wedge ... \wedge D_n,$
$C_1 \wedge C_2 \wedge ... \wedge C_n \rightarrow A, ? \pm |C_1, C_2, ..., C_n|)$.

(A)

$?\mathbf{C}(a)$
$\mathbf{C}(a) \wedge \mathbf{F}(w_1, a, Q_1) \rightarrow \mathbf{U}(a, o_1, Q_1)$
$\mathbf{C}(a) \wedge \mathbf{F}(w_2, a, Q_2) \rightarrow \mathbf{U}(a, o_2, Q_2)$
...
$\mathbf{C}(a) \wedge \mathbf{F}(w_n, a, Q_n) \rightarrow \mathbf{U}(a, o_n, Q_n)$
$\mathbf{F}(w_1, a, Q_1) \wedge ... \wedge \mathbf{F}(w_n, a, Q_n)$
$\mathbf{U}(a, o_1, Q_1) \wedge ... \wedge \mathbf{U}(a, o_n, Q_n \rightarrow \mathbf{C}(a))$
$? \pm |\mathbf{U}(a, o_1, Q_1), ..., \mathbf{U}(a, o_n, Q_n)|$
$?\mathbf{U}(a, o_1)$

- $\mathbf{U}(a, o_1, Q_1)$; $?\mathbf{U}(a, o_2, Q_2)$
  - $\mathbf{U}(a, o_2, Q_2)$; ...; $?\mathbf{U}(a, o_{n-1}, Q_{n-1})$
    - $\mathbf{U}(a, o_{n-1}, Q_{n-1})$; $?\mathbf{U}(a, o_n, Q_n)$
      - $\mathbf{U}(a, o_n, Q_n)$; $\mathbf{C}(a)$
      - $\neg\mathbf{U}(a, o_n, Q_n)$; $\neg\mathbf{C}(a)$
    - $\neg\mathbf{U}(a, o_{n-1}, Q_{n-1})$; $\neg\mathbf{C}(a)$
  - $\neg\mathbf{U}(a, o_2, Q_2)$; $\neg\mathbf{C}(a)$
- $\neg\mathbf{U}(a, o_1, Q_1)$; $\neg\mathbf{C}(a)$

Jest to oczywiście pewna słabość testu Turinga. Ma ona jednak swoje źródło w niejasności kryterium „bycia człowiekiem”, czy też „bycia inteligentnym” (w rozumieniu ludzkiej inteligencji). Problemy na tym polu doskonale odzwierciedlają dyskusje wokół zagadnienia adekwatności testu Turinga przedstawione w rozdziale drugim.

Możemy sobie również wyobrazić bardziej wyrafinowaną operacjonalizację. Załóżmy, że sędzia jako strategie w teście Turinga przyjmuje e-scenariusz zbudowany w oparciu o jeden z przedstawionych w tym podrozdziale schematów e-scenariuszy. Obrany e-scenariusz sędzia będzie wykorzystywał w grze jako metascenariusz. Ten metascenariusz wyraża początkową strategię sędziego. Sędzia przyjmuje jednak operacyjne kryteria uznawania, że odpowiednie „kryterium człowieczeństwa” (zawarte w przesłankach metascenariusza) jest/nie jest spełnione. Oznacza to, że zapytania metascenariusza nie zostaną zadane graczowi P. Sędzia dla każdego zapytania metascenariusza układa bowiem subscenariusz. Dopiero zapytania takiego subscenariusza (jako operacjonalizacje konkretnych zapytań metascenariusza sędziego) zadawane są testowanemu graczowi. Takie rozwiązanie umożliwia sędziemu uzyskanie o graczu P rzeczywistych informacji, które są użyteczne z punktu widzenia przeprowadzanego testu. Jest to możliwe dzięki konstrukcji subscenariuszy. Subscenariusze bowiem powinny być tak zbudowane, aby sprawdzać rzeczywistą wiedzę/umiejętności gracza P, nie zaś jego deklaracje.

Korzystając z przytoczonego już przykładu z umiejętnością odpowiadania na pytania subkognitywne, można powiedzieć, że pytanie:

— „Czy P potrafi odpowiadać na pytania subkognitywne?”

stanowiło będzie jedno z zapytań metascenariusza. Jego operacjonalizacją będzie natomiast subscenariusz zawierający takie zapytania, jak:

— „Czy słowo *Flugly* nadaje się na pseudonim artystyczny gwiazdy Hollywood?”
— „Czy słowa *Flugly* można użyć jako imienia misia przytulanki?”
— „Czy można użyć orzechów kokosowych jako instrumentów muzycznych?”
— „Czy świeżo ścięta trawa ładnie pachnie?”

Oczywiście każdy z subscenariuszy w swoich przesłankach reprezentuje odpowiednie przekonania sędziego odnośnie do tego, jaka odpowiedź na poszczególne zapytania będzie satysfakcjonująca. Sędzia może dołączyć subscenariusze do metascenariusza dzięki operacji wklejania.

Taka forma operacjonalizacji wydaje się zgodna z intuicjami przedstawionymi przez S. Watta we wspomnianym już artykule „Can People Think? Or Machines? A Unified Protocol for Turing Testing” [Watt 2009]. Skonstruowany przez niego szkic uniwersalnego protokołu przeprowadzania testu Turinga można by z powodzeniem uznać za podstawę dla początkowego metascenariusza sędziego. Poprawnie przygotowany protokół przeprowadzania TT stanowiłby swego rodzaju gwarancję trafnej identyfikacji gracza w teście Turinga.

Przedstawiony model jest oczywiście ujęciem bardzo wstępnym, opierającym się w dużej mierze na sugestii A. M. Turinga mówiącej o tym, że warto najpierw rozważyć test Turinga z pytaniami rozstrzygnięcia. Wydaje się jednak, że już na tym poziomie TT_{IEL} jest użyteczny dla analiz testu Turinga. Jego atrakcyjność polega na naturalnym, moim zdaniem, ujęciu testu Turinga jako systemu pytań i odpowiedzi oraz zastosowaniu do jego zbadania inferencyjnej logiki pytań. Formułując TT_{IEL} starałem się oddać z dużą dokładnością oryginalne warunki przeprowadzania TT, skupiając się na perspektywie sędziego w TT (która była dotychczas zaniedbywana w literaturze przedmiotu). Celem takiego podejścia jest analiza TT na pewnym poziomie ogólności (co umożliwia wykorzystanie narzędzi formalnych) przy jednoczesnym zachowaniu intuicji związanych z testem. Ze wstępnych analiz wynika, że wykorzystując pragmatyczną interpretację e-scenariuszy możemy je potraktować jako zapis strategii dla sędziego w TT. Wydaje się również, że wybór tak skonstruowanej strategii jest bardzo korzystny dla sędziego z powodu pewnych własności, które posiadają e-scenariusze. TT_{IEL} potwierdza również przypuszczenia N. Blocka, że dla przebiegu testu Turinga kluczowe znaczenie ma zasób wiedzy i przekonania, którymi kieruje się sędzia w TT.

Rozdział 4

Test Turinga — inspirująca gra

We wstępie do niniejszej książki podałem długą listę dyscyplin, na których test Turinga odcisnął swoje piętno. Znalazły się na niej zarówno filozofia, jak i psychologia, a także informatyka (por. s. 8). Gdy uwzględnimy dodatkowo fakt, że tematyka związana z testem Turinga cieszy się nadal dużym zainteresowaniem, możemy zaryzykować twierdzenie, że jest to jedna z najbardziej inspirujących znanych nam gier.

Przedstawię tutaj pewne koncepcje oraz rozwiązania, które stanowią twórcze rozwinięcie idei zawartych w teście Turinga. Rozdział ten podzielony jest na dwie części. Pierwsza z nich zawiera omówienie ciekawych propozycji zastąpienia testu Turinga innym testem (który lepiej spełniałby zadanie wyznaczone przez Turinga). W drugiej części przedstawiam pewną klasę systemów służących automatycznej autoryzacji użytkownika. Ich główne założenia bezpośrednio odwołują się do testu Turinga.

4.1. Wybrane propozycje alternatywne względem testu Turinga

W rozdziale 2 przedstawiłem pewne próby wzmocnienia bądź osłabienia oryginalnej propozycji Turinga. Część badaczy skłania się jednak ku zastąpieniu TT zupełnie nową propozycją, która będzie lepszym kryterium badania inteligencji systemów sztucznych. Opiszę tutaj trzy takie propozycje: *Inverted Turing Test* (odwrócony TT), *Lovelace Test* (test lady Lovelace) oraz *Minimum Intelligent Signal Test* (MIST).

4.1.1. Odwrócony test Turinga (*Inverted TT*)

S. Watt zaproponował swój test w artykule „Naive Psychology and the Inverted Turing Test” [Watt 1996]. Konstruując odwrócony TT (*Inverted Turing Test* — ITT) oparł się on na idei psychologii naiwnej (*naive psycho-*

logy). Pojęcie psychologii naiwnej pojawiło się po raz pierwszy w artykule A. Clarka „From Folk Psychology to Naive Psychology" [Clark 1987]. W tym tekście Clark stara się wykazać, że dla zrozumienia codziennych ludzkich zachowań powinniśmy posłużyć się konstruktem innym, niż ten zaproponowany przez tak zwaną psychologię potoczną (*folk psychology*). Psychologia potoczna traktowana jest jako rodzaj prymitywnej teorii, która ujawnia się jedynie w praktycznym działaniu (por. [Shanahan 1999]). Zdaniem Clarka powinniśmy porzucić próby wyjaśniania codziennych zachowań ludzi w świetle tak rozumianej teorii, czyli zespołu poglądów, przekonań żywionych przez tych ludzi. Stąd propozycja psychologii naiwnej, wedle której psychika jest zespołem kompetencji kognitywnych (por. [Clark 1987, s. 146]), które wykształciły się i ewoluowały wraz z rozwojem społecznym człowieka (gwarantując szybką i efektywną współpracę w grupie). To właśnie te kompetencje pozwalają na zrozumienie zachowań innych. Zdaniem S. Watta system dostrzega, rozpoznaje i reaguje na stany mentalne ludzi w taki sposób, w jaki robi to zwykły człowiek tylko wtedy, gdy kieruje się psychologią naiwną. Psychologia naiwna stanowi bowiem:

> [...] naturalną ludzką tendencję i zdolność do przypisywania stanów mentalnych innym i sobie samym — mówiąc krótko, umiejętność rozpoznawania i rozumienia innych umysłów [Watt 1996, s. 3].

Watt proponuje zatem przetestowanie maszyny pod względem zgodności jej zachowania z przewidywaniami psychologii naiwnej. Aby to osiągnąć, wystarczy — zdaniem Watta — zastąpić sędziego w klasycznym teście Turinga maszyną.

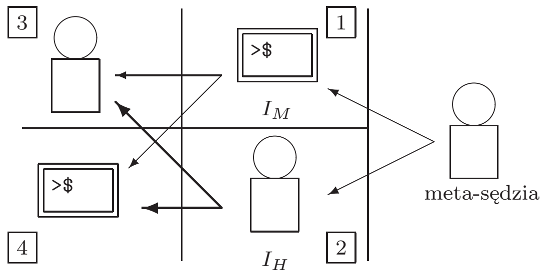
> Zamiast rozwijać zdolność systemów do zwodzenia ludzi, powinniśmy przeprowadzić test który sprawdzi, czy system przypisuje inteligencję innym w taki sam sposób jak robią to ludzie [Watt 1996, s. 6].

Mamy więc do czynienia z sytuacją, w której to sędzia (z klasycznej wersji TT) poddawany jest testowi. Watt podkreśla, że ITT ma charakter statystyczny. Test powinien być powtarzany wielokrotnie. Po każdej serii testów wyniki dla maszyny byłyby porównywane z wynikami uzyskanymi dla sędziego, który jest człowiekiem.

> Aby „zdać" odwrócony test Turinga system musi podlegać tym samym prawidłowościom i anomaliom w przypisywaniu stanów mentalnych, którym podlegałaby osoba — regularności i anomalie, które mogą być zbadane przez psychologię na zasadzie porównania [Watt 1996, s. 7].

Pomysł odwróconego testu Turinga skrytykował R. French w artykule „The Inverted Turing Test: How a Mindless Program Could Pass It" [French 1996]. Głównym punktem, na którym opiera on swoją krytykę, jest fakt, że ITT może być symulowany przy zastosowaniu układu znanego z klasycznego testu Turinga. Rysunek 4.1 ilustruje właśnie taki system.

Pomysł polega na wprowadzeniu tzw. meta-sędziego. Umieszczamy sędziów: człowieka (I_H) oraz maszynę (I_M) w oddzielnych pokojach (pokoje 1 i 2 na rysunku). Meta-sędzia nie wie, w którym z pokojów jest I_H, a w którym I_M. Sędziowie I_M oraz I_H oceniają graczy znajdujących się w kolejnych



Rysunek 4.1. Symulacja odwróconego testu Turinga przy użyciu klasycznej wersji testu Turinga (na podstawie [French 1996]); wyjaśnienia w tekście

pokojach (3 i 4). I_M oraz I_H poddawani są tym samym odwróconemu testowi Turinga. Meta-sędzia wie, kto jest w pokojach 3 i 4. Przeprowadza on klasyczny TT, przepytując jedynie sędziów I_H oraz I_M. Meta-sędzia zleca im następujące zadanie: „Wyobraźcie sobie, że jesteście sędziami w teście Turinga. Powiedzcie mi, na podstawie odpytywania osób z pokojów 3 i 4, czy macie do czynienia z maszyną i człowiekiem, czy może z dwojgiem ludzi." Na podstawie odpowiedzi udzielanych przez sędziów I_H i I_M meta-sędzia musi teraz ustalić, który z nich jest człowiekiem, a który maszyną. Jeżeli werdykty wydane przez I_M oraz I_H będą z perspektywy meta-sędziego nierozróżnialne, znaczyło to będzie, że maszyna (czyli sędzia I_M) zdała test.

W zaprezentowanym układzie ITT boryka się — zdaniem Frencha — z takimi samymi problemami, co oryginalny TT. ITT ograniczony jest bowiem do badania funkcji behawioralnych. Możemy więc wyobrazić sobie sytuację, w której sędzia będący maszyną (I_M) oszuka meta-sędziego. Zdaniem Frencha możliwe jest opracowanie tzw. *Human Subcognitive Profile* — HSP, który prezentuje sposób udzielania odpowiedzi na pewien zestaw pytań subkognitywinych (por. s. 41) uzyskany z dużej próbki (np. dzięki grze w ocenianie — *rating game*). Załóżmy teraz, że w ITT zarówno I_H, jak i I_M będą stosowali tylko pytania subkognitywne, przy czym sędziego-maszynę wyposażymy w zestaw tych pytań oraz uzyskany dla tego zestawu HSP. I_M zadaje po prostu pytania z posiadanej listy, po czym porównuje odpowiedzi z tymi zawartymi w HSP. Tym samym — zdaniem Frencha — maszyna będzie osiągała takie same wyniki, jak sędzia-człowiek (odpytujący z własnej listy pytań subkognitywnych). French podkreśla przy tym, że efekt taki osiągnięto dzięki prostemu wybiegowi.

4.1.2. Test lady Lovelace (*Lovelace Test*)

Inną propozycję zastąpienia TT, jaką jest test lady Lovelace (*Lovelace Test* — LT), znajdujemy w artykule „Creativity, the Turing Test, and the (Better) Lovelace Test" [Bringsjord et al. 2001]. Jego autorzy zwracają uwagę na pewien problem zauważony już przez Turinga w „Computing Machinery...",

a określany jako zarzut lady Lovelace (*Lady Lovelace's objection*). Lady Lovelace miała zanotować o maszynie analitycznej (*Analytical Engine*) skonstruowanej przez Charlesa Babbage'a:

> Maszyna analityczna nie rości sobie pretensji do *tworzenia* czegokolwiek. Potrafi jedynie wykonywać to, *o czym wiemy* jak *nakazać* jej to wykonać [Turing 1950, s. 450].

Innymi słowy, komputer nie jest w stanie stworzyć niczego sam z siebie — wykonuje jedynie program. Opierając się na tej własności maszyn cyfrowych, autorzy wspomnianego tekstu projektują swój test, który — ich zdaniem — lepiej niż TT nadaje się do badania inteligencji sztucznych systemów poznawczych.

Warunki pozytywnego przejścia LT są dokładnie określone (por. [Bringsjord et al. 2001, s. 12]).

Zaprojektowany przez H sztuczny system poznawczy A, dający na wyjściu wyniki $o_1, o_2, ..., o_n$*, zdaje LT* wtedy i tylko wtedy gdy:

(i) Istnieje przynajmniej jeden wynik o_k $(1 \leq k \leq n)$, co do którego H, lub ktoś o wiedzy H (oraz dysponujący równymi jemu zasobami technologicznymi i ludzkimi), odwołując się do bazy wiedzy, architektury i bazowych funkcji A nie potrafi wyjaśnić, w jaki sposób A uzyskał wynik o_k, oraz

(ii) Wynik o_k nie jest rezultatem błędu, ale procesu, który A jest w stanie powtórzyć.

W omawianym artykule znajdziemy również próbę doprecyzowania kwestii długości trwania testu oraz tego, jakimi zasobami i zakresem wiedzy może dysponować H. Jeśli chodzi o czas potrzebny H (do udzielenia wyjaśnienia wyniku uzyskanego przez A), to powinien mieć „[...] tyle czasu ile on, lub ona uważają za słuszne [...]" [Bringsjord et al. 2001, s. 9]. Oczywiście istnieje zdroworozsądkowa granica oczekiwania na wyjaśnienie ze strony H, za którą przyjmuje się w cytowanym artykule „kilka lat". Co do zasobów i wiedzy H, to powinien on dysponować wiedzą dotyczącą budowy testowanego sztucznego systemu poznawczego (czyli wiedzą na temat jego bazy wiedzy oraz sposobów implementacji jego głównych funkcji). H powinien również dysponować zasobami technologicznymi i ludzkimi, które pozwolą mu na zbadanie wymienionych elementów.

Przykładem zadania, które można wykorzystać w LT jest gra o nazwie L^3G (*The Short Story Game*). Gra ta polega na tym, że maszyna i człowiek dostają proste zdanie (np. cytat z *Przebudzenia* F. Kafki: „Gdy Gregor Samsa obudził się pewnego rana z niespokojnych snów, stwierdził, że zmienił się w łóżku w potwornego robaka"[1]). Ich zadaniem jest napisanie krótkiego opowiadania (około 500 słów) nawiązującego do tego zdania. Prace oceniane są pod względem wartości literackich, spójności, nowatorskości etc. (por. [Bringsjord 2001] oraz [Bringsjord et al. 2001]). S. Bringsjord i D. Ferruci zaprojektowali system BRUTUS, który doskonale radzi sobie w tego typu

---

[1] F. Kafka, Przebudzenie, w: *Cztery opowiadania. List do ojca*, tłum. J. Kydryński, J. Ziółkowski, PIW, Warszawa 2003.

grze (por. opowiadania stworzone przez ten system — [Bringsjord 2001, s. 26] i [Bringsjord et al. 2001, s. 14]). Nie możemy jednak powiedzieć, że BRUTUS przechodzi pomyślnie LT, ponieważ jego autorzy są w stanie dokładnie wytłumaczyć, w jaki sposób system ten tworzy wspomniane opowiadania (ponieważ sami tworzyli i implementowali algorytmy, z których on korzysta). Tutaj właśnie tkwi atrakcyjność LT — nie wystarczy imitacja szeroko pojmowanych zachowań behawioralnych, sztuczny system poddawany testowi musi wykazać się kreatywnością.

Zdaniem S. Bringsjorda, którego zainteresowania badawcze skupiają się na zagadnieniu maszynowej twórczości, nie jest możliwe, aby jakikolwiek sztuczny system zdał test Lady Lovelace.

> Prawdopodobnie nie jest możliwe aby artefakt przetwarzający jedynie informacje zdał LT, ponieważ to, czego szuka Lovelace wymagać może takiego rodzaju autonomii, która wykracza poza granice zwykłej relacji przyczynowej oraz matematyki [Bringsjord et al. 2001, s. 25].

4.1.3. MIST — *Minimum Intelligent Signal Test*

Minimum Intelligent Signal Test (MIST) został zaproponowany przez Chrisa McKinstry'ego w bardzo krótkim (zaledwie dwustronicowym) artykule „Minimum Intelligence Signal Test: an Objective Turing Test" [McKinstry 1997]. Bardziej dokładny opis proponowanego testu znaleźć można w późniejszym tekście opublikowanym w tomie *Parsing the Turing Test* (por. [McKinstry 2009].

Zdaniem Ch. McKinstry'ego głównym problemem związanym z testem Turinga jest fakt, że dzięki niemu możemy uzyskać wyłącznie jedną z dwóch (możliwych) odpowiedzi na pytanie, czy dana maszyna myśli — albo będzie to odpowiedź twierdząca, albo przecząca. McKinstry twierdzi, że TT nie dopuszcza żadnych pośrednich możliwości. Ujmuje to w sposób następujący:

> W teście Turinga dostajemy „wszystko albo nic" i przez to jest on bezużyteczny dla procesu tworzenia lub dokonywania pomiaru powstających systemów inteligentnych. Jedyne czego może nam dostarczyć, to informacji, że stworzyliśmy taki system (ale dopiero po fakcie). Tym czego naprawdę potrzebujemy jest test podobny do testu Turinga, w którym uznaje się pewną stopniowalność i traktuje inteligencję jako, co najmniej, pewnego rodzaju kontinuum zachowań ludzkich. Potrzebujemy testu, który pozwoliłby nam mierzyć to minimum globalnej ludzkiej inteligencji, które stanowi podstawę dla wykształcenia się dojrzałej inteligencji — testu, który mógłby zostać zautomatyzowany i dzięki temu byłby wykonywany z prędkością dostępną maszynom [McKinstry 2009, s. 286].

Aby osiągnąć tak zamierzony efekt McKinstry proponuje, aby MIST opierał się na porównaniu wzorców odpowiedzi udzielanych przez ludzi, z udzielonymi przez sztuczny system poznawczy. Z tego powodu w MIST powinny pojawiać się wyłącznie pytania i zdania oznajmujące, na które można odpowiadać „tak" lub „nie". Uważa on, że dzięki temu możliwe jest skoncentrowanie uwagi na wykrywaniu rzeczywistych, inteligentnych wzorców zachowań — a tym samym uniknięcie problemów, z jakimi borykają się organizatorzy

konkursu Lobenera. W przypadku konkursu Loebnera (a także oryginalnego TT) złożoność reakcji na bodźce testowe może wprowadzić sędziego w błąd. Ponadto praca sędziego nie daje się w nich zautomatyzować — a przez to wynik testu może być nietrafny z powodu niedoskonałości sędziego-człowieka (por. rozdział 3). W MIST odpowiedzi na bodźce testowe są skrajnie ograniczone, co ma gwarantować, że „kandydaci [testowani — przyp. P.Ł.] nie mogą udzielać odpowiedzi wymijających, mogą jedynie odpowiadać w sposób, w jaki odpowiadają ludzie lub nie" [McKinstry 2009, s. 289]. Zdaniem McKinstry'ego do wykorzystania w MIST doskonale nadają się zadania zaproponowane przez R. Frencha w jego „grze w ocenianie" (por. [French 1990] oraz rozdział 2.3.1.2), np. „Czy »Flugly« byłoby dobrym nazwiskiem aktorki?", a także zdania typu: „Istnieję", „Jesteś skałą", „Nie jesteś człowiekiem" (por. [McKinstry 2009, s. 290]).

Kolejnym, kluczowym elementem propozycji McKinstry'ego jest wykorzystanie pojęcia prawdopodobieństwa w ocenianiu wyników MIST, a także przyjęcie, że wynik testu będzie informował jedynie o stopniu podobieństwa między wzorcem inteligentnych odpowiedzi udzielanych przez ludzi a wzorcem odpowiedzi uzyskanym od maszyny. Procedurę testowania wyobraża sobie McKinstry w sposób następujący (por. [McKinstry 1997, s. 17], [McKinstry 2009, s. 288]):

1. *Generowane jest N bodźców testowych (pytań i zdań oznajmujących).* Bodźce te powinny być tak ułożone, aby ludzie (czy też bardziej realistycznie — pewna populacja ludzi) byli w stanie odpowiednio na nie zareagować. Dodatkowo — o czym była mowa powyżej — przyjmuje się, że reakcje te powinny mieć „binarny" charakter (czyli sprowadzać się do potwierdzenia lub zaprzeczenia). Na około 50% bodźców ludzie powinni reagować twierdzeniem, a na drugą część zaprzeczeniem. W tym kroku zbiera się również odpowiedzi na przygotowane bodźce testowe (w tym celu powinno zaangażować się jak najliczniejszą populację ludzi). W efekcie otrzymujemy bazę zawierającą dane zorganizowane na zasadzie bodziec testowy — odpowiedź, która posłuży ocenie wyników MIST.
2. Przygotowane bodźce testowe prezentuje się testowanemu podmiotowi w losowej kolejności i rejestruje się jego odpowiedzi.
3. Ocenia się zestaw zebranych odpowiedzi na bodźce testowe, porównując je z danymi zebranymi w pierwszym etapie procedury. McKinstry sugeruje, aby ten proces zautomatyzować, dzięki czemu uniknie się stronniczości i błędów ze strony sędziego porównującego dwa zestawy zebranych danych.
4. Im większy procent zgodności odpowiedzi udzielonych przez testowany system poznawczy z odpowiedziami zebranymi w pierwszym kroku, tym lepszy wynik tego podmiotu w MIST. McKinstry przyjmuje, że zgodność na poziomie 50% mogą osiągnąć podmioty odpowiadające w sposób losowy. Tym samym interesujące wyniki powinny być znacznie wyższe od tej dolnej granicy.

McKinstry w 2000 roku rozpoczął prace nad *Mindpixel Digital Mind Modeling Project*. Celem było stworzenie systemu, który mógłby przeprowadzić MIST. Niestety, projekt został wstrzymany po śmierci jego autora w 2006.

Zarówno odwrócony test Turinga, jak i test lady Lovelace oraz MIST stanowią bardzo interesujące alternatywy dla testu Turinga. Warto jednak zauważyć, że omówione propozycje również należałoby uznać za testy wejścia/wyjścia (por. [Crockett 1994]). W tym sensie nawiązują one bezpośrednio do idei testu Turinga — proponują pewne zawężone kryterium posiadania inteligencji. Dodatkowo owo kryterium bazuje na porównaniu działań (czy też raczej efektów działań) takiego systemu z podobnymi działaniami wykonywanymi przez ludzi. Dzięki temu ITT, LT oraz MIST dają nadzieję na praktyczną realizację, w odróżnieniu od propozycji skomplikowania testu Turinga opisanych w rozdziale drugim.

4.2. Praktyczna realizacja idei TT — systemy CAPTCHA

4.2.1. Systemy CAPTCHA — charakterystyka

Powszechny dostęp do globalnej sieci Internet stanowi niewątpliwie krok naprzód w dziedzinie komunikacji, jednocześnie jednak stwarza równie ogromny problem zapewnienia bezpieczeństwa owej komunikacji. Jedna z głównych zalet Internetu — szeroko zakrojona automatyzacja komunikacji — jest również jednym z najbardziej oczywistych i najgoręcej dyskutowanych zagrożeń: korzystając z Internetu, chcielibyśmy mieć pewność, że dane, które przesyłamy lub udostępniamy w globalnej sieci, nie zostaną wykorzystane przez osoby do tego niepowołane. Dotyczy to przede wszystkim następujących dziedzin wymiany i udostępniania danych w sieci WWW (por. [Bergmair, Katzenbeisser 2004], [Chew, Baird 2003], [Naor 1996], [Rui, Liu 2004]):

— *Darmowe konta e-mailowe.* Specjalnie skonstruowane programy (boty) rejestrują tysiące darmowych kont, aby później rozsyłać z nich *spam* lub *wirusy komputerowe.*
— *Serwisy udostępniające darmowe usługi.* Część firm oferuje nieodpłatne korzystanie z pewnych usług, np. wyszukiwarek, ofert, katalogów, a także z meta-usług, takich jak programy umożliwiające porównywanie cen u różnych producentów. Dostarczenie tego typu usług jest dla firmy kosztowne, ale spełnia swój konkretny marketingowy cel: przyciąga i wiąże klienta z daną firmą. Sytuacja ta zmienia się, gdy ktoś wykorzystuje — dostarczane nieodpłatnie — dane i usługi dla własnych korzyści.
— *Serwisy, w których liczy się liczba przeprowadzanych przez użytkownika transakcji, wizyt itp.* Jeżeli pozycja i uprawnienia użytkownika zależą od pewnego wskaźnika, którym jest, powiedzmy, liczba wizyt na określonej

stronie WWW, to chcielibyśmy mieć pewność, że użytkownik ten jest konkretną osobą, nie zaś botem.

— *Wszelkiego rodzaju głosowania online.* Aby zapewnić wiarygodność wyników musimy zagwarantować, że głosowali będą ludzie, a nie skonstruowane dla tego celu automaty[2].
— *Prywatność i ograniczony dostęp do danych.* Serwisy oferujące możliwość zabezpieczenia danych *loginem* i *hasłem* powinny mieć możliwość rozpoznania prób skorzystania z takich danych przez niepowołanego użytkownika.

Jak łatwo zauważyć, wymienione zagadnienia wiążą się z problemem automatycznego rozpoznania czy system ma do czynienia z człowiekiem, czy ze specjalnie skonstruowanym programem komputerowym. Taki właśnie cel stawia się przed systemami określanymi jako CAPTCHA (*Completely Automatic Public Turing Test to Tell Computers and Humans Apart*).

Systemy CAPTCHA są szczególną klasą protokołów określanych nazwą *Human Interactive Proofs* (HIP). Protokół HIP można najogólniej scharakteryzować jako taki rodzaj dowodu interakcyjnego (por. [Goldwasser et al. 1985], [Papadimitriou 2002]), który człowiek może z łatwością skonstruować, zaś maszyna nie może tego zrobić z równą łatwością (por. [Chew, Baird 2003]). Z tego właśnie powodu protokoły HIP są szczególnie przydatne w sytuacji, kiedy potrzebujemy przyjaznego użytkownikowi narzędzia autoryzacji, pozwalającego dodatkowo na odrzucanie wszelkich prób autoryzacji ze strony programów komputerowych. Do tego celu wykorzystuje się właśnie systemy CAPTCHA.

Za „ojca” CAPTCHA uważa się Moni Naora. W opublikowanym jedynie w Internecie tekście „Verification of a Human in the Loop or Identification via the Turing Test” jako pierwszy zaproponował on wykorzystanie idei testu Turinga oraz trudnych problemów z dziedziny sztucznej inteligencji do automatycznego rozpoznawania użytkowników-ludzi[3].

Już sama nazwa: *Completely Automatic Public Turing Test to Tell Computers and Humans Apart* wskazuje na podstawowe cechy tej klasy protokołów HIP. Są one *zautomatyzowane*, ponieważ mają być przeprowadzane przez komputer wyposażony w specjalny program komputerowy. Określane są mianem testu Turinga, ponieważ ich celem jest trafne zidentyfikowanie użytkownika, który jest człowiekiem. CAPTCHA zakorzenione są więc w idei TT. Zwraca się jednak uwagę, że — podczas gdy TT oparty jest na dialogu

---

[2] W tym kontekście często przytacza się przykład głosowania zorganizowanego w 1999 r. przez serwis http://slashdot.com. Była to forma rankingu pod hasłem: „Która uczelnia kształci najlepszych informatyków?”. Każdy mógł oddać głos na swoją uczelnię (zwyciężała ta z nich, która zbierze ich najwięcej). Już pierwszego dnia studenci Carnegie Mellon University napisali program głosujący po tysiąckroć na ich uczelnię. Następnego dnia to samo zrobili studenci z MIT. Głosowanie przerodziło się w zmagania botów. Końcowy wynik to 21 156 głosów oddanych na MIT, 21 032 głosy oddane na CMU (podczas gdy na inne uczelnie oddano — w tradycyjny sposób — poniżej 1 000 głosów).

[3] Co ciekawe, nazwa CAPTCHA pojawiła się dopiero później. M. Naor nazywa swoją propozycję po prostu „zautomatyzowanym testem Turinga”. Interesujący nas rodzaj systemów bywa również nazywany „odwrócony test Turinga” (*reverse Turing test*) — por. [Kochanski et al. 2002].

— CAPTCHA mogą mieć również inny charakter. Najogólniej CAPTCHA określa się następująco (por. [Ahn et al. 2003, s. 3], [von Ahn et al. 2008]): CAPTCHA jest protokołem kryptograficznym, którego konstrukcja oparta jest na pewnym trudnym problemie z dziedziny sztucznej inteligencji (SI)[4].

Nie każdy problem z dziedziny SI nadaje się do wykorzystania w dziedzinie bezpieczeństwa. Potrzebny jest bowiem sposób na zautomatyzowanie procesu generowania kolejnych wersji zadań testowych. Co więcej, gotowy system musi być tak przyjazny użytkownikowi, jak to tylko możliwe (zważywszy na praktyczne wykorzystanie systemów CAPTCHA). Przez przyjazność dla użytkownika rozumie się tutaj głównie takie skonstruowanie zadań testowych, aby ich rozwiązanie nie wymagało odwoływania się do wiedzy fachowej, a także aby wymagało ono minimalnych nakładów pracy. Zazwyczaj przyjmuje się, że poziom, w jakim dany system CAPTCHA realizuje wymienione postulaty znajduje swoje odzwierciedlenie w czasie rozwiązywania poszczególnych zadań testowych, ich poprawności, a także liczbie powtórzeń dla pojedynczego użytkownika. Zwraca się również uwagę na składane przez użytkowników deklaracje dotyczące np. subiektywnego poczucia trudności zadań. Warto podkreślić, że zagadnienie użyteczności systemów CAPTCHA nabiera coraz większego znaczenia w obliczu rosnącej liczby ich zastosowań w Internecie (por. np. [May 2005], [Yan, El Ahmad 2008*b*], [Vora 2009]).

Ponadto na CAPTCHA nakłada się warunek *publiczności*, który oznacza, że zarówno kod programu, jak i dane, które ów program wykorzystuje powinny być publicznie dostępne. Motywacją jest zapewnienie większego bezpieczeństwa — CAPTCHA musi być tak skonstruowany, żeby nawet przy publicznym dostępie do informacji na jego temat nie można było napisać programu, który go złamie. Jak łatwo zauważyć, kluczowe jest tu dobranie odpowiednio trudnego problemu z dziedziny SI (por. [Ahn et al. 2004]).

Jako najistotniejsze wymienia się następujące cechy systemów CAPTCHA (por. [Naor 1996, s. 2]; [Baird et al. 2003, s. 159]):

1. Kolejne zadania testowe są generowane automatycznie.
2. Test może zostać szybko rozwiązany przez użytkownika, który jest człowiekiem.
3. Test przechodzą jedynie ludzie (bez konieczności odwoływania się do specjalistycznej wiedzy).
4. Testu nie są w stanie zdać automaty (boty).
5. Test nie deaktualizuje się w obliczu zmian w technologii i metodach programistycznych (pomimo publicznego dostępu do algorytmów owego testu)[5].

Zwraca się również uwagę na fakt, że rolę sędziego w CAPTCHA spełnia nie człowiek, jak w TT, ale maszyna. Na pierwszy rzut oka stanowi to

---

[4] Formalne ujęcie problematyki CAPTCHA — sformułowane w ramach teorii dowodów interakcyjnych — zainteresowany czytelnik znajdzie w [Ahn et al. 2003].

[5] To założenie jest oczywiście nieco idealizacyjne. Systemy CAPTCHA opierają się bowiem na trudnych problemach z dziedziny SI, ale są to problemy, z jakimi boryka się ta dyscyplina na chwilę obecną, a które w przyszłości mogą zostać rozwiązane — por. np. algorytm Mori i Malik opisany w tym rozdziale.

nawiązanie do idei odwróconego testu Turinga autorstwa S. Watta (por. rozdział 4.1.1). Nie powinniśmy jednak zapominać o tym, że ów sędzia (będący programem przeprowadzającym test) posiada wiedzę (czy też może raczej umiejętność) dostępną jedynie człowiekowi, ale zaimplementowaną w programie w celu automatyzacji przeprowadzania testu. Analogia z odwróconym testem Turinga jest więc tutaj jedynie powierzchowna. Istotną różnicą pomiędzy CAPTCHA a TT jest to, że TT składa się z całej serii zadań stawianych przez sędziego maszynie, zaś w przypadku CAPTCHA — z racji ich przeznaczenia użytkowego — sędzia prezentuje jedno zadanie i tylko na podstawie niego musi zadecydować czy ma do czynienia z człowiekiem, czy z botem.

M. Naor w tekście, w którym zawarł ideę CAPTCHA, wymienił kilka problemów z dziedziny SI, które jego zdaniem mogłyby służyć jako podstawa do skonstruowania systemu CAPTCHA (por. [Naor 1996, s. 2–3]):

1. Rozpoznawanie płci. Zadaniem użytkownika byłoby tutaj określenie płci osoby na zdjęciu.
2. Rozpoznawanie wyrazu twarzy. Na podstawie zdjęcia twarzy należy określić, w jakim nastroju jest przedstawiona osoba.
3. Odnajdywanie części ciała. Na zdjęciu (przedstawiającym na przykład zwierzę) należałoby wskazać kliknięciem myszki część ciała, o wskazanie której proszony jest użytkownik.
4. Określanie nagości. Na dwóch zdjęciach prezentowane są osoby. Użytkownik ma rozpoznać, która z nich jest rozebrana.
5. Rozpoznawanie schematycznych rysunków. Użytkownikowi prezentuje się bardzo schematyczny rysunek (np. rysunek autorstwa dziecka). Zadanie polega na wybraniu jednej z pięciu nazw określających, co jest na rysunku.
6. Rozpoznawanie pisma odręcznego. Przedstawione słowo napisane ręcznie (z dodanym tłem) należy wpisać w przygotowane pole.
7. Rozpoznawanie mowy. Zadaniem użytkownika jest rozpoznanie nagranych słów i wpisanie ich w przygotowane pola.
8. Praca ze zdaniami. Jedną z propozycji jest, aby zadaniem użytkownika było poskładanie zdania z porozrzucanych słów. Drugą, aby test polegał na uzupełnianiu zdania wyrazem z listy.
9. Radzenie sobie z wieloznacznością słów. Oto przykład: zadanie testowe składałoby się z dwóch zdań, np. „The dog killed the cat. It was taken to the morgue”. Zadaniem użytkownika jest określenie do czego odnosi się słówko „it” w drugim zdaniu.

Realizacji doczekało się niewiele z tych propozycji, co zapewne spowodowane jest wymaganiami stawianymi przed systemami CAPTCHA. Systemy te, po pierwsze, muszą być bezpieczne, ale nade wszystko — i to jest czynnikiem decydującym o ich atrakcyjności — muszą być przyjazne użytkownikowi.

4.2.1.1. Rozpoznawanie obrazu (OCR CAPTCHA)

Jednym z problemów z zakresu SI, który najlepiej (jak dotąd) sprawdzał się jako podstawa do projektowania systemów CAPTCHA jest OCR (*optical character recognition*), szczególnie zaś problem rozpoznawania obiektu w tle (*object recognition in scenes*) (por. [Mori, Malik 2003]). Zazwyczaj tego rodzaju systemy CAPTCHA generują obrazek, zawierający napisy umieszczone na w pewien sposób zakłóconym tle. Zadaniem poddawanego testowi jest rozpoznanie owych słów i wpisanie ich w przygotowane wcześniej pole. Innym podejściem jest wykorzystanie obrazków lub zdjęć. Od użytkownika wymaga się tutaj rozpoznania tego, co przedstawia dana ilustracja lub jej szczegół. Omówię tu pokrótce wybrane popularne propozycje CAPTCHA wykorzystujące problem OCR: *PessimalPrint*, *Gimpy*, *BaffleText*, *Pix*, *ARTiFACIAL*.

PessimalPrint. System ten został zaproponowany w [Baird et al. 2003]. Tworząc kolejne zadania testowe, system wybiera losowo:

— słowo (z ustalonej listy),
— parametry czcionki: krój, rozmiar, styl (z ustalonej listy),
— zbiór metod zniekształcania (ze z góry określonego zakresu metod, do którego należą m. in. rozmywanie oraz progowanie[6]).

Korzystając z tych trzech elementów, system generuje jeden czarno-biały obrazek. System dysponuje listą 70 angielskich słów, o długości od 5 do 8 liter. Słowa wybrane są spośród najczęściej używanych w Internecie. Zbiór stosowanych zniekształceń jest dobrany ze względu na problemy, z jakimi nie radzą sobie systemy OCR. Najogólniej mówiąc, celem zastosowania zniekształceń jest imitowanie fizycznych zniszczeń tekstu, które powstają podczas kopiowania lub skanowania tekstów. Stosowane metody zniekształceń to na przykład:

— zwężanie obrazu, tak aby powstało wrażenie zlewania się poszczególnych znaków,
— dodawanie szumu (rozmaitych zakłóceń) do obrazu,
— użycie czcionek o wąskich krojach,
— użycie czcionek o pochylonych krojach.

Przykładowe zadanie wygenerowane przez system *PessimalPrint* przedstawia rysunek 4.2.

Autorzy *PessimalPrint* przeprowadzili testy systemu z wykorzystaniem programów OCR (Expervision TR, ABBYY FineReader oraz IRIS Reader). Wyniki tych testów przedstawiają się bardzo obiecująco: z 685 wygenerowanych wyrazów, Expervision TR w całości rozpoznał tylko 0,29% z nich, zaś pozostałe dwa programy nie rozpoznały żadnego ze słów. Niestety, autorzy nie przeprowadzili żadnych eksperymentów z udziałem ludzi, które mogłyby wykazać, że — pomimo dużych zniekształceń — tekst jest dla nich łatwo czytelny. Stanowi to warunek praktycznego wykorzystania sytemu CAPTCHA (na co wskazują losy opisanego poniżej systemu *Gimpy*).

[6] Progowanie (*thresholding*) służy do konwersji rysunku do czerni i bieli (bez wykorzystania odcieni szarości).



Rysunek 4.2. Przykładowe zadania wygenerowane przez system *PessimalPrint* (por. [Baird et al. 2003])

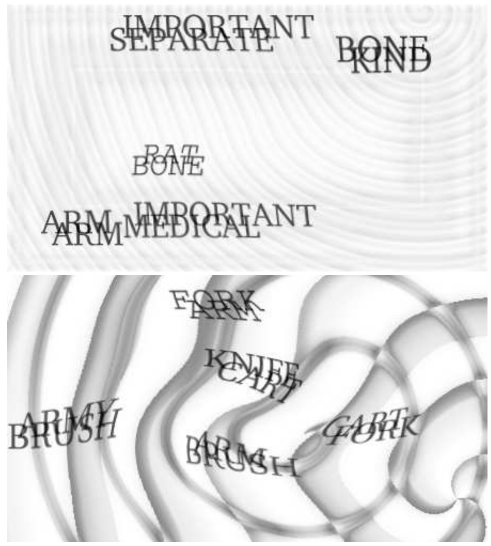
Gimpy. Ten system CAPTCHA również opiera się na wprowadzaniu zniekształceń do tekstu. *Gimpy* dysponuje słownikiem składającym się z 850 angielskich słów. System korzysta z programu Gimp oraz zestawu czcionek Free Type. Używane są różne rodzaje zniekształceń:

— manipulowanie tłem (dodawanie siatki tła, gradientu),
— rozmywanie,
— dodawanie szumu,
— deformacje kształtu tekstu.

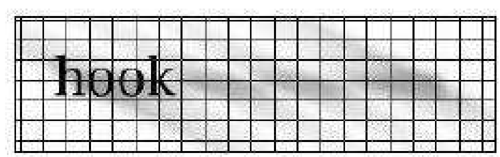
System wybiera 7 słów ze swojego słownika i wykorzystuje je do przygotowania obrazka testowego. Użytkownik musi rozpoznać trzy z siedmiu zaprezentowanych słów. Rysunek 4.3 przedstawia przykłady testów wygenerowanych przez system *Gimpy*.

Gimpy był przez krótki czas używany przez portal Yahoo! Okazało się jednak, że jest on oceniany przez użytkowników jako zbyt trudny. *Gimpy* zastąpiony został łatwiejszym CAPTCHA — *EZ Gimpy*, w którym wymaga się rozpoznania tylko jednego słowa (por. rysunek 4.4). Stanowi to doskonały przykład tego, że systemy CAPTCHA powinny spełniać dwa kryteria — dla maszyn powinny być trudne, ale jednocześnie powinny być jak najłatwiejsze i jak najmniej kłopotliwe dla użytkownika. Tutaj uwidacznia się wyzwanie, jakie stoi przed projektantami systemów CAPTCHA, którzy muszą pogodzić ze sobą oba te warunki.

BaffleText. Zaproponowany w [Chew, Baird 2003], stanowi pewną innowację w stosunku do CAPTCHA typu *Gimpy* czy *PessimalPrint*. Autorzy *BaffleText* zdecydowali się bowiem na wykorzystanie ciągów znaków, które nie są angielskimi słowami (warunkiem jest to, żeby owe ciągi znaków dawały się w miarę łatwo wymówić). Motywacją takiego kroku była zbyt duża przewidywalność poprzednich systemów, wynikająca z małego słownika słów angielskich (często wybieranych spośród słów najczęściej pojawiających się



Rysunek 4.3. Przykłady testów wygenerowanych przez system *Gimpy*. Użytkownik musi rozpoznać trzy z siedmiu zaprezentowanych na obrazku słów. Źródło: `www.captcha.net`



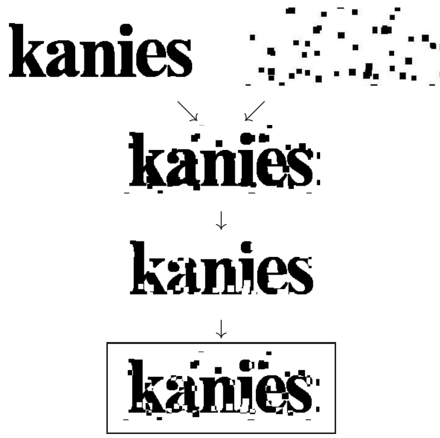
Rysunek 4.4. Przykładowe zadanie *EZ Gimpy*. Źródło: `www.captcha.net`

w Internecie). Procedura tworzenia testu *BaffleText* jest następująca (por. [Chew, Baird 2003, s. 5]):

1. Wygenerowanie ciągu znaków, który nie jest angielskim słowem (ale bardzo przypomina jakieś słowo), np.: *obviouse*, *alued*, *emperly*, *magine*, *ourses*, *thates* (5 do 8 znaków).
2. Wybranie jednej z wielu dostępnych czcionek.
3. Wygenerowanie ciągu znaków przy użyciu wybranej czcionki — utworzenie w rezultacie obrazka.
4. Wygenerowanie maski obrazu (czyli dodatkowej warstwy obrazu).
5. Wybranie jednej z operacji obróbki maski obrazu.
6. Połączenie warstw obrazu (ciągu znaków i maski).

Rysunek 4.5 przedstawia etapy tworzenia zadania poprzez dodawanie zakłóceń do obrazka ze słowem testowym.

Decyzja o wykorzystywaniu ciągów znaków zamiast rzeczywistych słów jest niewątpliwie dużą zaletą systemu (ze względu na poziom bezpieczeństwa), ale i jego największą słabością. Podobieństwo „słów” testowych do rzeczywistych słów może powodować liczne pomyłki. Powodem może być tutaj wskazywany przez psychologię poznawczą efekt przewagi słowa nad literami



Rysunek 4.5. Etapy tworzenia przykładowego testu BaffleText [Chew, Baird 2003]

(*word superiority effect*). Zjawisko to przejawia się na przykład w podświadomym uzupełnianiu brakujących liter w wyrazie, a nawet całych wyrazów, których brakuje w zdaniu, tak aby uzyskać sensowną całość (por. [Nęcka et al. 2006, s. 316–317]). Autorzy *BaffleText* przeprowadzili badania na 33 osobach (pracownikach firmy PARC). Badani rozwiązali w sumie 1212 testów. 79% z nich zostało rozwiązanych poprawnie. Mierzono również średni czas rozwiązania zadań. Dla odpowiedzi poprawnych wynosił on $6,6$ sekundy, zaś dla odpowiedzi niepoprawnych był wyraźnie dłuższy i wynosił 15 sekund. Wydaje się, że to właśnie efekt przewagi słowa nad literami jest odpowiedzialny za stosunkowo długi czas rozwiązania zadania (należało przecież rozpoznać tylko jedno słowo)[7]. Ciekawym elementem badania był formularz badający akceptację użytkownika dla systemu *BaffleText* (wypełniło go 18 z 33 osób badanych). Pytania koncentrowały się wokół deklarowanej chęci używania systemu. I tak:

— 3 osoby wyraziły chęć rozwiązywania testu *BaffleText* każdorazowo, kiedy wysyłają pocztę elektroniczną.
— 7 osób wyraziło chęć rozwiązywania testu *BaffleText* każdorazowo, kiedy wysyłają pocztę elektroniczną, pod warunkiem, że dziesięciokrotnie zmniejszy to liczbę przychodzącego *spamu*.
— 16 osób wyraziło chęć rozwiązywania testu *BaffleText* każdorazowo przy rejestrowaniu się na stronie internetowej związanej z handlem elektronicznym (*e-commerce*).

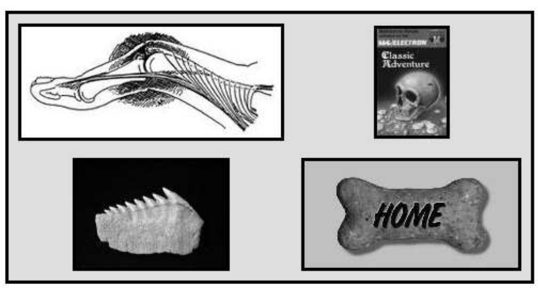
[7] Pewien wpływ na długość rozwiązania zadania mógł mieć czas wpisywania rozwiązania w przygotowane do tego celu pole. Taki pomiar czasu w sytuacji badania nie jest zbyt korzystny, ponieważ nie mówi nam ile tak naprawdę zajęło badanemu odczytywanie znaków. M. Chew i H. S. Baird niestety nie podają w swoim tekście tego, czy „czas rozwiązania" jest łącznym czasem rozwiązania zadania (na które składa się również wpisywanie rozpoznanych znaków), czy etap wpisywania został w jakiś sposób odjęty.

— 17 osób wyraziło chęć rozwiązywania testu *BaffleText* każdorazowo przy rejestrowaniu się na stronie internetowej wymagającej podania poufnych informacji.
— 18 osób wyraziło chęć rozwiązywania testu *BaffleText* każdorazowo przy rejestrowaniu darmowego konta poczty elektronicznej.

Badania tego typu są z pewnością bardzo potrzebne w kontekście wszystkich systemów CAPTCHA (obok sprawdzania, jak z rozwiązywaniem testów radzą sobie ludzie i programy). Pozwalają one bowiem na przynajmniej częściową odpowiedź na pytanie o przyjazność projektowanego systemu dla użytkownika.

Część badań nad systemami CAPTCHA skupia się raczej na rozpoznawaniu pewnych obiektów niż słów. Motywacją jest tutaj zapewne większa łatwość rozwiązywania takich zadań z perspektywy użytkownika, który jest człowiekiem. Omówimy tutaj dwie propozycje tego typu: system *ESP-PIX* oraz system *ARTiFACIAL* opierający się na unikatowej zdolności człowieka do rozpoznawania twarzy (por. [Nęcka et al. 2006, s. 313–314]).

ESP-PIX. W CAPTCHA *ESP-PIX* zadanie polega na rozpoznaniu obiektu wspólnego dla wyświetlanych obrazków. Przykład takiego zadania przedstawia rysunek 4.6. System dysponuje dużą bazą etykietowanych obrazków, które przedstawiają jakiś konkretny przedmiot. Generowanie zadania polega na losowym wybraniu i wyświetleniu czterech obrazków, na których znajduje się ten sam przedmiot. System pyta użytkownika „o czym są te obrazki". Odpowiedź należy wybrać z przygotowanej listy (w celu zminimalizowania możliwości pojawienia się wieloznaczności odpowiedzi).



Rysunek 4.6. Przykład zadania wygenerowanego przez system *ESP-PIX*. Źródło: www.captcha.net

Pewną modyfikację systemu *ESP-PIX* stanowi *Animal–Pix*. Wykorzystuje się tutaj zniekształcone zdjęcia (por. rysunek 4.7) dwudziestu zwierząt (m.in. niedźwiedzia, krowy, psa, słonia, konia, kangura, lwa, małpy, świni i węża). Użytkownik proszony jest o wybranie jednej z przygotowanych etykiet dla wyświetlanego obrazka.

Systemy typu *ESP-PIX* są bardzo przyjazne użytkownikowi. Ich rozwiązanie jest łatwe i nie zajmuje dużo czasu. Pewnym problemem jest jednak ich efektywna implementacja. Zagrożeniem dla bezpieczeństwa systemu jest

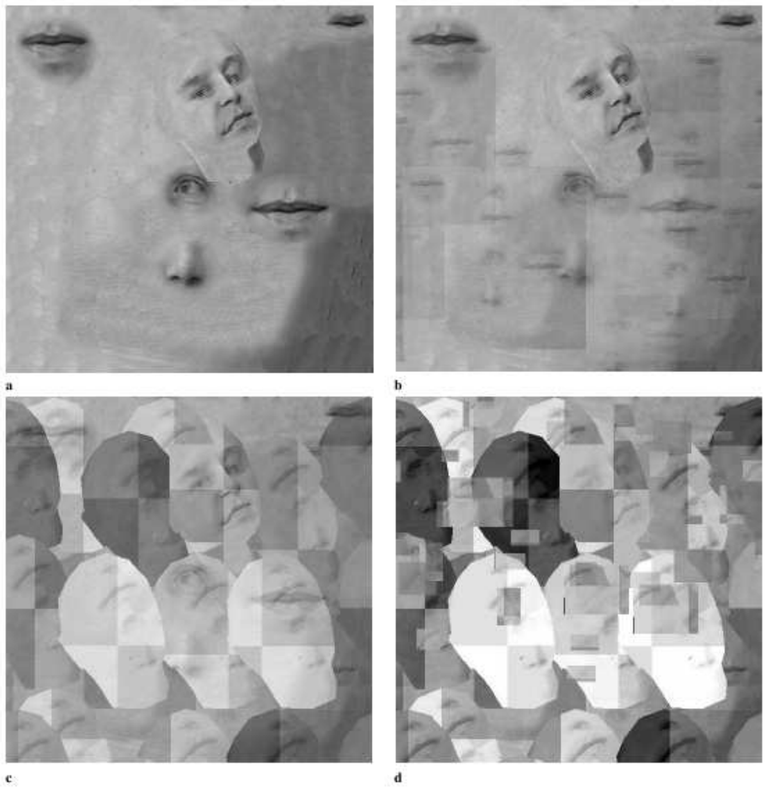


Rysunek 4.7. Przykład testu Animal–Pix [Ahn et al. 2003]

tutaj konieczność etykietowania każdego z obrazków w bazie danych. Można więc sobie wyobrazić napisanie programu, który będzie rozwiązywał zadania testowe generowane przez *ESP-PIX* w oparciu o bazę danych systemu (która zgodnie z zasadami CAPTCHA powinna być ogólnodostępna). W pewnym stopniu zabezpieczeniem jest wprowadzenie losowych zniekształceń wybieranych każdorazowo obrazków (tak jak w *Animal–Pix*), aby utrudnić ich rozpoznanie systemom OCR (por. [Ahn et al. 2004, s. 59]).

ARTiFACIAL. System zaproponowany w [Rui, Liu 2004]. *ARTiFACIAL* oznacza *Automated Reverse Turing test using FACIAL features*. Zadaniem użytkownika jest rozpoznanie na obrazku testowym twarzy oraz wskazanie na niej sześciu punktów: lewego i prawego kącika obu oczu oraz lewego i prawego kącika ust. System generuje zadanie testowe w kilku krokach (por. rysunek 4.8). Najpierw — wykorzystując trójwymiarowy model ludzkiej twarzy — przygotowuje obraz zawierający pewne wybrane elementy twarzy (I_1 oraz I_2). Kolejnym krokiem jest wygenerowanie tła z powtarzających się kopii zniekształconej twarzy (I_3). W ostatnim kroku łączy się wszystkie obrazy w jeden, który będzie zadaniem testowym. Każde takie zadanie zawiera tylko jedną twarz przedstawioną w całości.

Autorzy *ARTiFACIAL* przeprowadzili próbne ataki na swój system, wykorzystując systemy automatycznego rozpoznawania twarzy oraz systemy rozpoznające jej elementy. Wyniki osiągnięte przez programy nie były imponujące. Na przykład system rozpoznawania elementów twarzy — na 1000 zaprezentowanych zadań testowych (przy założeniu, że twarz została już wskazana na obrazku) — rozpoznał wszystkie sześć punktów do wskazania jedynie w dwóch przypadkach! Zadania *ARTiFACIAL* są więc niewątpliwie trudne dla maszyn, ale czy są jednocześnie łatwe dla człowieka? Y. Rui i Z. Liu przeprowadzili również badania mające odpowiedzieć na to pytanie. Wzięły w nim udział 34 osoby, z których każda rozwiązała po 10 testów. Średni czas rozwiązania poszczególnych zadań testowych wynosił od 11 do 22 sekund. Zwraca uwagę stosunkowo długi czas rozwiązania zadania. Wynika on niewątpliwie z wymagań, jakie stawia *ARTiFACIAL* użytkownikowi. Najpierw musi wydobyć twarz z bardzo mylącego tła, a później wskazać na niej aż sześć punktów. Tło zastosowane w *ARTiFACIAL* może stanowić źródło wielu pomyłek. W [Nęcka et al. 2006] znajdziemy opisy eksperymentów z zakresu psycholo-



Rysunek 4.8. Kolejne fazy generowania testu ARTiFACIAL: (a) I_1, (b) I_2, (c) I_3, (d) gotowy test [Rui, Liu 2004]

gii poznawczej, w których najpierw prezentowano badanym całą twarz do identyfikacji, później jej fragmenty oraz twarze zniekształcone (np. z oczami umieszczonymi na wysokości ust, z minimalnie zmienionymi proporcjami twarzy). Zaobserwowano, że w drugim przypadku badani mają bardzo duże problemy z rozpoznaniem nawet znanych sobie twarzy (por. [Nęcka et al. 2006, s. 314], por. też [Stanford, Web 2006, s. 344–345]). Tym bardziej zaskakujące są wyniki uzyskane w badaniach *ARTiFACIAL*: na 340 testów badani błędnie rozpoznali twarz tylko w jednym z nich. Niestety Y. Rui i Z. Liu nie publikują w swoim artykule kompletu obrazków testowych, a jedynie ten z błędnie zidentyfikowaną twarzą, nie ma więc żadnej możliwości porównania ich trudności. Problemem w systemie *ARTiFACIAL* jest konieczność dopuszczenia drobnych błędów w kroku drugim (to znaczy we wskazywaniu sześciu punktów na odnalezionej twarzy). Badani wskazywali bowiem zadane punkty z pewną dozą niedokładności, co zapewne wynikało z zakłóceń wprowadzanych przez tło.

Mimo obiecujących wyników wydaje się, że praktyczne wykorzystanie systemu *ARTiFACIAL* może być kłopotliwe. Świadczy o tym poziom skomplikowania zadań (znajdujący swoje odzwierciedlenie w przedstawionych przez

autorów średnich czasach rozwiązania zadań). Przeciętny użytkownik może uznać za zbyt pracochłonne zadanie, polegające na rozpoznaniu twarzy i wskazywaniu na niej aż sześciu punków. Wystarczy porównać trudność zadań generowanych przez *Gimpy* (wskazywane przez użytkowników jako zbyt trudne) z obrazkami testowymi *ARTiFACIAL*, aby przekonać się, że te drugie są znacznie bardziej wymagające.

4.2.1.2. Inne typy CAPTCHA

Oczywiście systemy CAPTCHA, oparte na problemie OCR, nie wyczerpują całej gamy możliwości konstrukcji tego typu testów. CAPTCHA może być również zaprojektowany w oparciu o inne otwarte problemy z dziedziny sztucznej inteligencji.

Jednym z nurtów badań jest próba stworzenia systemu CAPTCHA opartego jedynie na tekście (główną motywacją jest tu wygoda i użyteczność wdrażania takiego testu do praktycznych zastosowań). Przykładem takiego systemu może być CAPTCHA autorstwa R. Bergmaira oraz S. Katzenbeissera [Bergmair, Katzenbeisser 2004]. System ten wykorzystuje zjawisko wieloznaczności pewnych słów w zależności od kontekstu. Zadanie polega na wybraniu ze zbioru zdań tych, które możemy sobą zastępować bez utraty sensu. Przykładowe zadanie wygenerowane przez ten system przedstawia rysunek 4.9. Warto zaznaczyć, że tekst prezentowany jest bez żadnych dodanych zakłóceń czy zniekształceń (inaczej niż w przypadku CAPTCHA opartych na problemie OCR).

Pick the sentences that are meaningful replacements of each other:

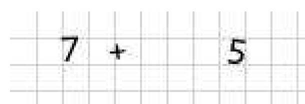
— The speech has to move through several more drafts.
— The speech has to run through several more drafts.
— The speech has to go through several more drafts.
— The speech has to impress through several more drafts.
— The speech has to strike through several more drafts.

Rysunek 4.9. Przykład tekstowego CAPTCHA [Bergmair, Katzenbeisser 2004]

Podejmowane są również próby stworzenia dźwiękowych systemów CAPTCHA (por. [Ahn et al. 2004, s. 59–60], [Kochanski et al. 2002]). Idea pozostaje podobna jak w przypadku testów opartych na rozpoznawaniu obrazu, ale tym razem mamy do czynienia z próbką dźwiękową, na którą nakładana jest „maska" z innych dźwięków (np. losowo odtwarzanych wybranych słów z danego słownika).

Łatwo zauważyć, że CAPTCHA tego typu są znacznie trudniejsze z perspektywy użytkownika niż te oparte na OCR. Tekstowe CAPTCHA wymagają od użytkownika stosunkowo dużego nakładu pracy i często — tak jak w przypadku systemu opisanego w [Bergmair, Katzenbeisser 2004] — dosko-

nałej znajomości języka. W przypadku dźwiękowych CAPTCHA konieczne jest posiadanie odpowiedniego sprzętu (choćby słuchawek lub głośników). Są to z pewnością powody tego, że dotychczas największą popularnością cieszą się CAPTCHA oparte na OCR, zaś pozostałe stanowią zazwyczaj propozycje teoretyczne. Stanowi to motywację dla nurtu badań mających na celu poprawienie bezpieczeństwa CAPTCHA opartych na problemie OCR bez zwiększania stopnia ich trudności dla użytkowników-ludzi. Jedną z rozważanych możliwości jest dodanie do problemu OCR jakiegoś dodatkowego zadania, które nie utrudni rozwiązania ludziom, ale sprawi, że całość będzie bardziej problematyczna dla botów. W tym nurcie tworzone są na przykład tzw. „Math CAPTCHA". W CAPTCHA tego typu obrazek testowy zawiera zadanie matematyczne. Użytkownik musi więc najpierw rozpoznać tekst, a następnie rozwiązać zadanie matematyczne i podać wynik. Przykład takiego zadania (pozyskany z www.php-help.ro/examples/math_captcha_image/) przedstawiony jest na rysunku 4.10. Niestety, problem leżący u podłoża tego typu systemów CAPTCHA nie jest trudny dla maszyn. Samo odczytanie zdegradowanego obrazka nie stanowi zbyt dużego problemu, zaś rozwiązanie zadania polegającego np. na dodaniu dwóch liczb jest dla komputera trywialne (należy tu podkreślić, że w Math CAPTCHA nie mogą pojawić się bardziej wyrafinowane zadania matematyczne, ponieważ będą one zbyt kłopotliwe dla ludzi).



Rysunek 4.10. Przykładowe zadanie Math CAPTCHA

W tym kontekście naturalne wydaje się szukanie inspiracji dla systemów CAPTCHA wśród wyższych poziomów przetwarzania informacji przez ludzi. Jako przykłady takich systemów można wymienić między innymi:

— *ARTiFACIAL*, w którym wykorzystuje się ludzką łatwość rozpoznawania twarzy (por. [Rui, Liu 2004]).
— *ESP-PIX*, gdzie dla rozwiązania konieczne jest skojarzenie cechy wspólnej czterem obrazkom (por. [Ahn et al. 2003]).
— *Egglue CAPTCHA*, gdzie użytkownik musi uzupełnić zdanie, w którym brakuje czasownika, tak aby to zdanie miało sens[8].
— *SemCAPTCHA*, w którym, aby podać rozwiązanie, użytkownik musi najpierw rozpoznać trzy wyrazy, następnie odnaleźć wzorzec, zgodnie z którym zostały one dobrane i wskazać to, które nie pasuje do pozostałych (por. [Łupkowski, Urbański 2008*a*], [Łupkowski, Urbański 2008*b*]).

Problemy leżące u podłoża wymienionych systemów CAPTCHA (dodane niejako do problemu OCR) z całą pewnością nie są trywialne dla maszyn.

[8] Por. http://code.google.com/p/egglue/

Jak wskazują badania związane z tymi propozycjami systemów CAPTCHA, dodatkowe elementy implementowane w tych systemach nie stanowią jednocześnie zbyt dużego obciążenia dla użytkowników. Można więc zaryzykować twierdzenie, że wytyczają one nowe drogi rozwoju — tak obecnie popularnych — systemów CAPTCHA.

4.2.2. Dlaczego warto konstruować CAPTCHA?

Jednym z powodów, dla którego warto konstruować systemy CAPTCHA są zapewne korzyści praktyczne oraz szerokie spektrum zastosowań tych systemów opisane w rozdziale 4.2.1. W kontekście pytania, będącego tytułem tego podrozdziału, wskazuje się jednak częściej na pozytywny wpływ systemów CAPTCHA na rozwój badań z zakresu sztucznej inteligencji. W [Ahn et al. 2003] znajdziemy następującą opinię na ten temat:

> Ważną składową sukcesu współczesnej kryptografii jest przyjęta na jej gruncie praktyka formułowania w sposób bardzo przejrzysty i jasny warunków, przy których możemy uznać pewne protokoły kryptograficzne za bezpieczne. Praktyka ta pozwala wspólnocie uczonych na ewaluację owych założeń i próby ich złamania. W przypadku sztucznej inteligencji, bardzo rzadko zdarza się, żeby problemy były tak precyzyjnie sformułowane, ale wykorzystanie ich do celów bezpieczeństwa wymusza na projektantach protokołów ową precyzję sformułowań. Jesteśmy przekonani, że precyzyjne ujęcie nierozwiązanych problemów SI może przyspieszyć rozwój sztucznej inteligencji [...] [Ahn et al. 2003, s. 295].

Przykładem mogą tu być CAPTCHA oparte na problemie OCR. Ich odczuwalna obecność w Internecie wymusiła niejako próby złamania tego typu zabezpieczeń. Powstało wiele prac, których autorzy prezentują techniki i programy rozwiązujące tego typu zadania. Warto tu wspomnieć choćby o kilku z nich. Mori i Malik w artykule „Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA" [Mori, Malik 2003] opisują swój atak na CAPTCHA *Gimpy* i *EZ-Gimpy*. W pierwszym przypadku udało im się poprawnie rozwiązać 33% zadań testowych, w drugim aż 92%. G. Moy i współpracownicy [Moy et al. 2004] osiągnęli 99% poprawnie rozwiązanych zadań dla systemu *Gimpy*. Z kolei Yan i Ahmad [Yan, El Ahmad 2008*a*] opisali prostą metodę segmentacji, która umożliwiła złamianie CAPTCHA używanego przez Microsoft (60% poprawnie rozwiązanych zadań). Powstają również prace, w których przedstawiane są wyniki zastosowania bardzo prostych technik analizy obrazu do łamania systemów CAPTCHA opartych na OCR (por. m.in. [Yan, El Ahmad 2007], [Yan, El Ahmad 2009], [Łupkowski 2009]). Ciekawy jest również fakt, że obok wspomnianych prac, których celem jest wykazanie nieskuteczności CAPTCHA opartych na problemie OCR, zaczęto również zwracać uwagę na zagadnienia związane z użytecznością tych systemów (por. np. [May 2005], [Yan, El Ahmad 2008*b*], [Vora 2009]).

Oczywiście badacze zajmujący się projektowaniem systemów CAPTCHA nie pozostają w tyle. Na rynku CAPTCHA pojawiają się coraz to nowe rozwiązania, które częstokroć czerpią inspiracje z dziedzin wychodzących

poza OCR. Jedną z zasadniczych cech CAPTCHA jest to, że test pozostaje aktualny, pomimo zmian w technologii i metodach programistycznych oraz publicznego dostępu do algorytmów owego testu. Takie założenie wręcz prowokuje swoistą grę w „kotka i myszkę” pomiędzy twórcami CAPTCHA a badaczami ulepszającymi algorytmy rozwiązujące otwarte problemy SI. CAPTCHA stanowią więc pewnego rodzaju motywację do działania i okazję do precyzowania problemów, z jakimi boryka się dyscyplina określana mianem sztucznej inteligencji. Dobrze zaprojektowany (czyli precyzyjnie ujmujący pewien otwarty problem) system CAPTCHA generuje prostą sytuację: albo CAPTCHA nie zostaje złamany (i wtedy może służyć jako zautomatyzowany test Turinga), albo zostaje złamany, a tym samym zostaje rozwiązany otwarty problem, na którym system ten został oparty.

Aneks

Alan Mathison Turing (1912–1954)

Rodzice Alana Turinga — Julius Mathison oraz Ethel — przez większą część swojego dorosłego życia mieszkali w południowych Indiach. Julius Mathison był bowiem urzędnikiem Indyjskiej Służby Cywilnej. W Indiach urodził się starszy brat Alana — John (w 1908 roku). Rodzice zdecydowali jednak, że ich drugie dziecko powinno przyjść na świat na Wyspach Brytyjskich. I tak Alan Mathison Turing urodził się 23 czerwca 1912 roku w Londynie. Jeszcze w tym samym roku rodzice Alana powrócili do Indii pozostawiając obu synów pod opieką państwa Ward (zamieszkujących miejscowość St Leonards-on-Sea nieopodal Hastings). Julius M. Turing dopiero w 1929 roku przeszedł na emeryturę i na stałe osiadł na Wyspach Brytyjskich.

W 1926 roku Alan został przyjęty do publicznej szkoły Sherborne w Dorset. Po jej ukończeniu został przyjęty w roku 1931 do King's College w Cambridge. Studia ukończył z wyróżnieniem w 1934. W 1935 otrzymał stypendium King's College. Prawdopodobnie to wykłady z podstaw matematyki prowadzone w 1935 roku przez M. H. A. Newmana sprawiły, że skupił on swoją szczególną uwagę na zagadnieniu rozstrzygalności (por. [Hodges 1992, s. 90], [Hodges 1998, s. 12] [Hodges 2003]). Efektem podjętych prac był artykuł o ogromnym znaczeniu w dorobku naukowym A. M. Turinga: „On computable numbers, with an application to the *Entscheidungsproblem*", który ukazał się w *Proceedings of the London Mathematical Society* w roku 1937 (został on wysłany do publikacji w 1936 r.)[1]. To właśnie w tej pracy Turing formułuje ideę abstrakcyjnej maszyny znanej dziś jako Maszyna Turinga (MT)[2].

MT możemy sobie wyobrazić jako maszynę posiadającą — podzieloną na komórki — *taśmę* (nieskończoną w prawo[3]) oraz *głowicę*, która może w dowolnej chwili obserwować tylko jedną komórkę taśmy. Każda z komórek może zawierać tylko jeden ze skończonej liczby symboli taśmowych. Działanie MT

---

[1] Warto zaznaczyć, że w 1936 r. wyniki o podobnej wymowie ogłosił w pracy „An unsolvable problem in elementary number theory" Alonzo Church (por. s. 45 oraz 47).

[2] Maszyna Turinga była jednym z kilku równoważnych modeli obliczania zdefiniowanych w tamtym okresie (por. [Papadimitriou 2002, s. 68]).

[3] Tym terminem wyraża się fakt, że taśma MT posiada swój początek (skrajną lewą komórkę) ale — potencjalnie — nie posiada skrajnej prawej komórki.

określone jest przez „program" nazywany *sterowaniem skończonym*, na które składa się skończony zbiór stanów i zbiór przejść ze stanu do stanu, zachodzących przy różnych symbolach wejściowych odczytywanych z taśmy MT (por. [Hopcroft, Ullman 2003, s. 29]). Zgodnie z bieżącym stanem sterowania skończonego oraz symbolem obserwowanym przez głowicę taśmy MT może w pojedynczym ruchu (por. [Hopcroft, Ullman 2003, s. 173]): (1) zmienić stan, (2) wydrukować symbol w obserwowanej komórce taśmy, zastępując nim symbol uprzednio tam zapisany[4], (3) przesunąć głowicę o jedną komórkę w prawo, lewo lub pozostawić głowicę bez ruchu. Czytelnika zainteresowanego tematyką Maszyn Turinga odsyłamy do pozycji wymienionych w bibliografii, np. [Aho et al. 2003], [Hopcroft, Ullman 2003] lub [Papadimitriou 2002].

W „On Computable Numbers..." Turing wykorzystuje model obliczania, jakim jest Maszyna Turinga, do zmierzenia się z problemem sformułowanym przez Davida Hilberta — czy istnieje efektywna procedura (algorytm), która pozwoliłaby na rozstrzygnięcie zdania matematycznego (w tradycji niemieckojęzycznej problem ten znany jest właśnie jako *Endscheidungsproblem*). Turing udziela negatywnej odpowiedzi na pytanie o istnienie takiej procedury. Dziś powszechnie akceptowalnym modelem formalnym procedury efektywnej jest właśnie Maszyna Turinga (por. [Hopcroft, Ullman 2003, s. 172]).

W latach 1936–1938 Turing kontynuował swoje badania, przebywając na Uniwersytecie w Princeton w USA. Tam pod kierunkiem A. Churcha przygotował i obronił w czerwcu 1938 roku dysertację doktorską z logiki matematycznej, która rok później ukazała się w *Proceedings of the London Mathematical Society* pod tytułem „System of Logic Based on Ordinals" [Turing 1939].

W roku 1938 wraca do King's College w Cambridge. Warto wspomnieć, że w tym samym roku Turing uczęszczał na wykłady prowadzone w Cambridge przez Ludwiga Wittgensteina dotyczące podstaw matematyki[5]. W tym samym roku rozpoczyna również współpracę z Government Code and Cypher School (GC&CS).

Wraz z wybuchem drugiej wojny światowej Turing zostaje zatrudniony przez GC&CS w ośrodku Bletchely Park, gdzie ma zająć się problemem złamania kodu niemieckiej maszyny szyfrującej Enigma. Podobno Alan Turing był jednym z dwóch pracowników, którzy wierzyli w możliwość złamania kodu Enigmy (drugim był szef sekcji szyfrów marynarki Frank Brich). Brich uważał, że złamanie kodu Enigmy jest po prostu konieczne, tymczasem Turing chciał się po prostu zmierzyć z wyzwaniem, które powszechnie uważane było za bardzo trudne (por. [Oakley, Gallehawk 2007, s. 69]). Opierając się na wcześniejszych dokonaniach polskiego wywiadu oraz wynikach osiągniętych przez polskich kryptologów (szczególnie Mariana Rejewskiego, Jerzego Różyckiego i Henryka Zygalskiego) osiągnął on znaczne sukcesy na tym polu. Prawdopodobnie bazując na polskim projekcie urządzenia do automatyzacji

[4] Zakłada się tutaj istnienie symbolu pustego, którym oznaczone są niezapisane komórki taśmy.

[5] Atmosferę owych wykładów, prowadzonych raczej w formie grupy dyskusyjnej niż tradycyjnego wykładu uniwersyteckiego, doskonale oddaje książka Davida Edmondsa i Johna Eidinowa *Pogrzebacz Wittgensteina* [Edmonds, Eidinow 2002].

procesu dekryptażu depesz Enigmy (tzw. „bomba" autorstwa M. Rejewskiego) Turing zaprojektował elektryczno-mechaniczne urządzenie nazwane w Bletchley „pająkiem" (dzisiaj znane raczej jako „bomba Turinga" lub „bomba Turinga–Welchmana"). Szczegóły dotyczące konstrukcji Enigmy oraz opis zawiłej drogi wiodącej do złamania jej kodu zainteresowany Czytelnik znajdzie np. w [Oakley, Gallehawk 2007] i w [Karbowski 2006, rozdz. 1].



Rysunek 1. Pomnik Alana Turinga w Bletchely Park. Fot. Jon Callas

Już w trakcie prac w ośrodku w Bletchley Park Turing rozpoczął badania nad — jak to określał — inteligencją maszyn. Dziś moglibyśmy je z powodzeniem zaliczyć do pierwszych badań nad sztuczną inteligencją (por. [Copeland, Proudfoot 2009]). Niestety, szkic artykułu, który Turing przedstawił do dyskusji swoim współpracownikom zaginął.

W 1945 roku Alan Turing otrzymał Order Imperium Brytyjskiego (*Order of the British Empire*).

Po zakończeniu wojny Turing został zatrudniony w National Physical Laboratory (NLP) w Londynie. Z uwagi na jego doświadczenie nabyte w Bletchley, powierzone mu zostaje zadanie zaprojektowania pierwszego komputera cyfrowego ogólnego przeznaczenia (a więc maszyny, którą można programować). Z zadania tego Turing wywiązał się bardzo szybko, przedstawiając w ra-

porcie „Proposed electronic calclulator” z 1946 roku projekt komputera o nazwie *Automatic Computer Engine* (ACE)[6]. W literaturze przedmiotu zwraca się uwagę na fakt, że projekt Turinga był zadziwiająco szczegółowy (Turing oszacował nawet cenę zbudowania swojego komputera) i zawierał bardzo nowatorskie idee. Podkreśla się również to, że Turing już od samego początku prac nad swoim projektem twierdził, że dwa aspekty będą kluczowe dla każdej budowanej maszyny cyfrowej: prędkość działania i zasoby pamięciowe (por. [Copeland, Proudfoot 2000, s. 491]). Z przyczyn od Turinga niezależnych, jego oryginalny projekt nie został zrealizowany. Na jego podstawie w NPL skonstruowano jednak — między innymi — dostępny komercyjnie komputer DEUCE.

W trakcie pracy w NPL narasta jego frustracja związana z tym, że w tym czasie jego osiągnięcia z Bletchley pozostawały utajnione. Dodatkowo prace NPL zostają w tym czasie przyćmione przez amerykański projekt budowy komputera. W tym czasie Turing szuka odskoczni od pracy naukowej w uprawianiu sportu (biega w maratonach). W 1948 roku niemalże został zakwalifikowany do reprezentacji Wielkiej Brytanii na olimpiadę.

W maju 1948 roku otrzymał propozycję z Uniwersytetu w Manchesterze, którą przyjął. Pracował tam do końca swojej kariery akademickiej. Jego pierwszym zadaniem było opracowanie systemu programowania dla komputera Ferranti Mark I (komercyjnie dostępnego komputera produkowanego przez Uniwersytet w Manchesterze — sprzedano dziesięć sztuk tej maszyny, por. [Copeland, Proudfoot 2000, s. 492]). Ukończył je w 1951 roku. Warto nadmienić, że ulubionym problemem, który Turing rozważał w kontekście programowania komputerów, była gra w szachy. Turing zaczął nawet pisać program do gry w szachy — *Turochamp* — dedykowany dla komputera Ferranti Mark I. Co prawda nigdy go nie ukończył, ale sam program zasługuje na zainteresowanie z uwagi na wykorzystanie w nim różnego rodzaju heurystyk (co było w owym czasie nowatorskim podejściem do problemu).

W 1948 roku sporządził raport zatytułowany „Intelligent Machinery” (skan oryginalnego maszynopisu dostępny jest w *Digital Turing Archive* [Turing 1948]). Pierwsze zdanie tej pracy brzmi: „Proponuję rozważyć pytanie o to, czy jest możliwe aby maszyny przejawiały inteligentne zachowania” [Turing 1948, s. 1]. Raport ten postrzegany jest jako pierwszy manifest idei badań nad sztuczną inteligencją (por. [Copeland 2000], [Copeland, Proudfoot 2009]). W „Intelligent Machnery” Turing wprowadza wiele propozycji rozwiązań, które dzisiaj stanowią już rozwiązania klasyczne w ramach dyscypliny sztucznej inteligencji. Badacze wkładu Alana Turinga w rozwój tej dyscypliny — J. Copeland i D. Proudfoot — podkreślają, że znajdziemy w tym raporcie zagadnienia związane z koneksjonizmem, algorytmami genetycznymi oraz dowodzeniem twierdzeń (por. m.in. [Copeland, Proudfoot 2000, s. 495]).

W 1950 roku, w czasopiśmie *Mind* ukazuje się artykuł „Computing Machinery and Intelligence” zawierający propozycję znaną dzisiaj jako test Turinga. Jak komentuje A. Hodges:

---

[6] Skan oryginalnego raportu dostępny jest w *Turing Digital Archive* (AMT/C/32).

> Problem napisania tekstu przeznaczonego dla czytelników bez przygotowania matematycznego Turing rozwiązał z właściwą sobie zimną krwią, ignorując wszelkie konwencjonalne bariery kulturowe. Pozbawiony jakichkolwiek odwołań do literatury z dziedziny filozofii czy psychologii, artykuł Turinga jest bezkompromisowy zarówno pod względem stylu, jak i zawartości [Hodges 1998, s. 56].

Warto podkreślić, że również w tym artykule odnajdujemy bardzo nowatorską (jak na tamte czasy) propozycję skonstruowania uczących się maszyn (w paragrafie 7). Turing pisze:

> Zamiast próbować wyprodukować program symulujący umysł osoby dorosłej powinniśmy raczej spróbować wyprodukować taki program, który symulowałby dziecięcy umysł [...] Tym samym dzielimy problem na dwie części. Program-dziecko (*child-programme*) oraz proces nauczania. Obydwa pozostają ze sobą w ścisłym związku. Nie możemy oczekiwać że otrzymamy dobry progam-dziecko już przy pierwszej próbie. Należy eksperymentować z procesem nauczania takiej maszyny i obserwować jak się ona uczy. Następnie należy wypróbować inne metody, sprawdzając, czy są one lepsze czy gorsze [Turing 1950, s. 456].

W dalszej części artykułu Turing szegółowo opisuje i dyskutuje metody, jakich można by użyć w uczeniu maszynowym.

Tematyka związana z inteligentnymi maszynami zajmuje stałe miejsce w aktywności naukowej Turinga. Warto wspomnieć tu chociażby „Can Digital Computers Think" (wywiad radiowy wyemitowany w 1951 roku), „Intelligent Machinery, a Heretical Theory" (wykład wygłoszony w roku 1951), „Can automatic calculating machines be said to think?" (wywiad radiowy wyemitowany w 1952 roku) oraz „Digital Computers Applied to Games" (tekst z tomu *Faster than thought* pod redakcją B. V. Bowdena, który ukazał się w 1953 roku).

W 1951 roku Turing zostaje wybrany członkiem Królewskiej Akademii Nauk (*Fellow of the Royal Society*) w uznaniu wyników przedstawionych w „On Computable Numbers..."

W tym samym czasie zainteresowania Turinga zaczynają również oscylować wokół biologii i chemii. Interesuje go możliwość modelowania procesów biologicznych przy użyciu maszyn cyfrowych (wykorzystuje do tego komputer Ferranti Mark I). Wyniki swoich pionierskich prac prezentuje w artykule „The chemical basis of morphogenesis", który ukazał się w *Philosophical Transactions of the Royal Society of London* [Turing 1952].

Niestety, w wyniku nieszczęśliwego zbiegu okoliczności w 1952 roku został aresztowany i oskarżony o homoseksualizm (który w owym czasie był w Wielkiej Brytanii przestępstwem). Po przyznaniu się do „winy", został poddany przymusowej kuracji hormonalnej, która miała „wyleczyć" go ze „złych skłonności". Jednocześnie odsunięto go od wszystkich prac związanych z projektami rządowymi. Żyjąc w izolacji od świata naukowego, zmagał się ze skutkami kuracji hormonalnej. 7 czerwca 1954 roku w swoim domu w Wilmslow (Cheshire) popełnił samobójstwo, zjadając jabłko zatrute cyjankiem.

10 września 2009 roku premier Wielkiej Brytanii Gordon Brown — w odpowiedzi na kampanię społeczną zainspirowaną przez J. G. Cumminga — na łamach gazety *Telegraph* przeprosił za to, w jaki sposób został potraktowany Alan Turing. Napisał m. in.:

Był on jedną z tych postaci, których indywidualne zaangażowanie pomogło zmienić bieg wojny. Dług wdzięczności wobec niego sprawia, że nieludzki sposób, w jaki został potraktowany wydaje się być jeszcze bardziej przerażający. [...] Turing został skazany zgodnie z obowiązującym w tym czasie prawem i nie możemy cofnąć czasu, ale sposób, w jaki został potraktowany był oczywiście rażąco niesprawiedliwy. Dlatego też cieszę się, że mam szansę wyrazić to, jak bardzo mi przykro z powodu tego wszystkiego co go spotkało[7].

[7] Pełny tekst dostępny jest pod adresem http://tinyurl.com/37934qf.

Literatura

Ahn, L., Blum, M., Hopper, N. J., Langford, J. [2003], ‘CAPTCHA: Using Hard AI Problems For Security’, *Lecture Notes in Computer Science* **2656**, 294–311.

Ahn, L., Blum, M., Langford, J. [2004], ‘Telling humans and computers apart automatically. How lazy cryptographers do AI’, *Communications of the ACM* **47**(2), 57–60.

Aho, A. V., Hopcroft, J. E., Ullman, J. D. [2003], *Projektowanie i analiza algorytmów*, Helion, Gliwice. Przeł. W. Derechowski.

Baird, H. S., Coates, A. L., Fateman, R. J. [2003], ‘PessimalPrint: a reverse Turing test’, *International Journal on Document Analysis and Recognition* **5**, 158–163.

Belnap, N. D. [1969], Åqvist’s Corrections-Accumulating Question Sequences, *w:* J. W. Davis, D. J. Hockney, W. K. Wilson, (red.), ‘Handbook of Philosophical Logic. Second Edition’, Reidel, Dordrecht, s. 122–134.

Bergmair, R., Katzenbeisser, S. [2004], ‘Towards Human Interactive Poofs in the text-domain. Using the problem of sense-ambiguity for security’, *Lecture Notes in Computer Science* **3225**, 257–276.

Block, N. [1995*a*], The computer model of the mind, *w:* E. Smith, D. Osheron, (red.), ‘An Invitation to Cognitive Science — Thinking’, The MIT Press, London, s. 147–289.

Block, N. [1995*b*], The mind as the software of the brain, *w:* E. Smith, D. Osherson, (red.), ‘An Invitation to Cognitive Science — Thinking’, The MIT Press, Londyn, s. 377–425.

Bradford, P. G., Wollowski, M. [1995], ‘A formalization of the Turing Test’, *ACM SIGART Bulletin* **6**(4), 3–10.

Bringsjord, S. [2001], ‘Creativity, the Turing test, and the (better) Lovelace test’, *Mind and Machines* **11**, 3–27.

Bringsjord, S. [2009], If I Were Judge, *w:* G. B. R. Epstein, G. Roberts, (red.), ‘Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer’, Springer Publishing Company, Incorporated, rozdział 6, s. 89–102.

Bringsjord, S., Bello, P., Ferruci, D. [2001], ‘Chess is to easy’, *Technology Review* **101**(2), 23–28.

Čaplinskas, A. [1998], 'AI paradigms', *Journal of Intelligent Manufacturing* **9**, 493–502.

Caporael, L., Heyes, C. [1996], Why Anthropomorphize? Folk Psychology and Other Stories, *w:* R.W. Michell, N.S. Thompson, H.L. Miles, (red.), 'Anthropomorphism, Anecdotes, and Animals', University of New York Press, s. 59–73.

Casacuberta, D. [2007], *Umysł — czym jest i jak działa*, Świat Książki, Warszawa. Przeł. J. Krzyżanowski.

Chalmers, D.J. [1992], Subsymbolic computation and the Chinese Room, *w:* J. Dinsmore, (red.), 'The Symbolic and Connectionist Paradigms: Closing the Gap', Lawrence Erlbaum, Hillsdale, New Jersey Hove and London, s. 25–48.

Cherniak, C. [1988], 'Undebuggability and Cognitive Science', *Communications of the Association for Computing Machinery* **31**, 402–412.

Chew, M., Baird, H.S. [2003], Baffletext: A human interactive proof, *w:* 'Proceedings of SPIE-IS&T Electronic Imaging, Document Recognition and Retrieval X', s. 305–316.
URL: *http://citeseer.ist.psu.edu/chew03baffletext.html*

Chomsky, N. [2009], Turing on the "Imitation Game", *w:* G.B.R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 7, s. 103–106.

Churchland, P.M., Churchland, P. [1991], 'Czy maszyna może myśleć', *Świat Nauki*, s. 17–23.

Clark, A. [1987], 'From folk psychology to naive psychology', *Cognitive Science* **11**, 139–154.

Copeland, B.J. [2000], 'The Turing Test', *Mind and Machines* **10**, 519–539.

Copeland, B.J., Proudfoot, D. [2000], 'What Turing Did after He Invented the Universal Turing Machine', *Journal of Logic, Language, and Information* **9**, 491–509.

Copeland, J., Proudfoot, D. [2009], Turing's test: A philosophical and historical guide, *w:* G.B.R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 9, s. 119–138.

Copple, K.L. [2009], Bringing AI to Life: Putting Today's Tools and Resources to Work, *w:* G.B.R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 22, s. 359–376.

Crockett, L.J. [1994], *The Turing Test and the Frame Problem. AI's Mistaken Understanding of Intelligence*, Ablex Publishing Corporation, Nortwood, New Jersey.

Cullen, J. [2009], 'Imitation Versus Communication: Testing for Human-Like Intelligence', *Mind and Machines* **19**, 237–254.

Dalen, D. v. [2002], Algorithms and decision problems: A crash course in recursion theory, *w:* D.M. Gabbay, F. Guenthner, (red.), 'Handbook of

Philosophical Logic 2nd Edition', Vol. 1, Kluwer Academic Publishers, Netherlands, s. 245–311.

Damper, R.I. [2006], 'The logic of Searle's Chinese room argument', *Mind and Machines* **16**, 163–183.

De Angeli, A., Gerbino, W., Nodari, E., Petrelli, D. [1999], From tools to friends: Where is the borderline?, *w:* 'Proceedings of the UM'99 Workshop on Attitude, Personality and Emotions in User-Adapted Interaction', Canada, s. 1–10.

De Angeli, A., Graham, A., Johnson, I., Coventry, L. [2001], The Unfriendly User: Exploring Social Reactions to Chatterbots, *w:* 'Proceedings of the International Conference on Affective Human Factors Desing', Asean Academic Press, London.
URL: *http://www.alicebot.org/articles/guest/The Unfriendly User.pdf*

De Angeli, A., Lynch, P., Johnson, G. [2001], 'Personifying the e-market: A framework for social agents'. Proceedings of Interact 2001, July 9-11, Tokyo.
URL: *http://www.informatics.manchester.ac.uk/∼antonella/files/Pdf/Personifying the e-market framework for social agents.PDF*

Drozdek, A. [1998], 'Human Intelligence and Turing Test', *AI & Society* **12**, 315–321.

Dryer, D. C. [1999], 'Getting personal with computers: How to design personalities for agents', *Applied Artificial Intelligence* **13**(3), 273–295.

Edmonds, B. [2000], 'The constructiblity of artificial intelligence (as defined by Turing test)', *Journal of Logic, Language, and Information* **9**(4), 419–424.

Edmonds, D., Eidinow, J. [2002], *Pogrzebacz Wittgensteina*, Warszawskie Wydawnictwo Literackie MUZA, Warszawa. Przeł. L. Niedzielski.

Epstein, R., Roberts, G., Beber, G., (red.) [2009], *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Springer Publishing Company, Incorporated.

Erion, G. J. [2001], 'The Cartesian Test for Automatism', *Mind and Machines* **11**, 29–39.

French, R. [1990], 'Subcogniton and the Limits of the Turing Test', *Mind* **99**(393), 53–65.

French, R. [1996], 'The inverted Turing test: How a mindless program could pass it', *Psycholoquy* **7**(39).

French, R. [2000], 'The Turing Test: the First 50 Years', *Trends in Cognitive Sciences* **4**(3), 115–122.

Garner, R. [2009], The Turing Hub as a Standard for Turing Test Interfaces, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 19, s. 319–324.

Gelernter, D., (red.) [1994], *The muse in the machine: Computerizing the poetry of human thought*, NY Free Press, New York.

Genova, J. [1994], 'Turing's sexual guessing game', *Social epistemology* **8**(4), 313–314.

Goldwasser, S., Micali, S., Rackoff, C. [1985], The knowledge complexity of Interactive Proof Systems (extended abstract), *w:* 'Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC'85)', Providence Rhode Island, s. 291–304.

Gratch, J., Marsella, S. [2005], 'Lessons from emotion psychology for the design of lifelike characters', *Applied Artificial Intelligence* **19**(3–4), 215–233.

Gregory, R. [2000], *Mózg i maszyny*, Prognozy XXI wieku, Prószyński i S-ka, Warszawa. Przeł. L. Grobelski.

Gunderson, K. [1964], 'Descartes, La Mettrie, Language, and Machines', *Philosophy* **39**, 193–222.

Harnad, S. [2000], 'Mind, machines and Turing: the indistinguishability of indistinguishables', *Journal of Logic, Language, and Information* **9**, 423–455.

Harnish, R. M. [2002], *Minds, Brains, Computers. An Historical Introduction to the foundations of Cognitive Science*, Blackwell Publishers, Oxford.

Harrah, D. [1969], 'On Completeness in the Logic of Questions', *American Philosophical Qarterly* **6**(2), 158–164.

Harrah, D. [2002], The Logic of Questions, *w:* D. Gabbay, F. Guenthner, (red.), 'Handbook of Philosophical Logic. Second Edition', Kluwer, Dordrecht/Boston/London, s. 1–60.

Hauser, L. [1997], 'Searle's Chinese Box: Debunking the Chinese Room', *Mind and Machines* **7**, 199–266.

Hayes, P., Ford, K. [1995], Turing test considered harmful, *w:* 'Proceedings of the 1995 International Joint Conference on Artificial Intelligence (IJCAI 95)', Montreal, Quebeck, Canada.

Hernandez-Orallo, J. [2000], 'Beyond the Turing test', *Journal of Logic, Language, and Information* **9**, 447–466.

Hetmański, M. [2000], *Umysł a maszyny — krytyka obliczeniowej teorii umysłu*, Wydawnictwo UMCS, Lublin.

Hodges, A. [1992], *Alan Turing: The Enigma*, Vintage, London.

Hodges, A. [1998], *Turing*, Miniatury filozoficzne, Amber, Warszawa. Przeł. J. Nowotniak.

Hodges, A. [2003], Alan Turing, *w:* E. N. Zalta, (red.), 'The Stanford Encyclopedia of Philosophy', Stanford University.
URL: *http://plato.stanford.edu/*

Hopcroft, J. E., Ullman, J. D. [2003], *Wprowadzenie do teorii automatów, języków i obliczeń*, PWN, Warszawa. Przeł. B. Konikowska.

Horgan, J. [1999], *Koniec nauki, czyli o granicach wiedzy u schyłku ery naukowej*, Prószyński i S-ka, Warszawa. Przeł. M. Tempczyk.

Humphrys, M. [2009], How My Program Passed the Turing Test, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 15, s. 237–260.

Hutchens, J. L. [2009], Conversation simulation and sensible surprises, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophi-

cal and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 20, s. 325–342.

Jonas, H. [1996], *Zasada odpowiedzialności. Etyka dla cywilizacji technologicznej*, PLATAN, Kraków. Przeł. M. Klimowicz.

Karbowski, M. [2006], *Podstawy kryptografii*, Helion, Gliwice.

Kartezjusz [1637/1994], *Rozprawa o metodzie*, Morex, Warszawa. Przeł. T. Boy-Żeleński.

Kelly, A. [2003], *Decision Making Using Game Theory*, Cambridge University Press, Cambridge.

Kiepas, A. [1992], *Moralne wyzwania nauki i techniki*, Transformacje, Katowice.

Kloch, J. [1996], *Świadomość komputerów? Argument „Chińskiego Pokoju" w krytyce mocnej sztucznej inteligencji według Johna Searle'a*, Wydawnictwo OBI, Kraków.

Kochanski, G., Lopresti, D., Shih, C. [2002], A Reverse Turing Test Using Speech, *w:* 'Proceedings of ICSLP2002 (International Conference on Spoken Language Processing)', s. 1357–1360.

Konar, A. [2000], *Artificial Intelligence and Soft Computing. Behavioral and Cognitive Modeling of the Human Brain*, CRC Press, Boca Raton – London – N.Y. – Washington.

Krajewski, S. [2003], *Twierdzenie Gödla i jego interpretacje filozoficzne. Od mechanicyzmu do postmodernizmu*, Umysł — Prace z Filozofii i Kognitywistyki, Wydawnictwo IFiS PAN, Warszawa.

La Mettrie, J. O. [1748/1984], *Człowiek–maszyna*, PWN, Warszawa. Przeł. S. Rudniański.

Lassègue, J. [1996], 'What Kind of Turing Test Did Turing Have in Mind', *Tekhnema* **3**, 37–58.

Lassègue, J. [2009], Doing Justice to the Imitation Game: A Farewell to Formalism, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 11, s. 151–172.

Lem, S. [1984], *Dialogi*, Wydawnictwo Literackie, Kraków.

Lem, S. [1996*a*], *Summa technologiae*, Interart, Warszawa.

Lem, S. [1996*b*], *Tajemnica chińskiego pokoju*, Universitas, Kraków.

Lem, S. [1999], *Bomba megabitowa*, Wydawnictwo Literackie, Kraków.

Lewin, J., Gastiew, J., Rozanow, J. [1967], *Język, matematyka, cybernetyka*, PWN, Warszawa. Przeł. I. Roman.

Loebner, H. [2009], How to Hold a Turing Test Contest, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 12, s. 173–180.

Lucas, J. R. [1961], 'Minds, Machines and Gödel', *Philosophy* **XXXVI**, 112–127.

Luger, G. F., Stubblefield, W. A. [1998], *Artificial Intelligence. Structures and Strategies for Complex Problem Solving*, Addison Wesley Longman Inc.

Ławrow, I. A., Maksimowa, L. Ł. [2004], *Zadania z teorii mnogości, logiki matematycznej i teorii algorytmów*, Wydawnictwo Naukowe PAN, Warszawa. Przeł. J. Pogonowski.

Łupkowski, P. [2005*a*], Czy zagadnienie sztucznej inteligencji jest pseudoproblemem?, *w:* T. Mróz, M. Sieńko, (red.), 'Propositiones', Instytut Filozofii Uniwersytetu Zielonogórskiego, Zielona Góra, s. 61–71.

Łupkowski, P. [2005*b*], 'Rola etyki i antropologii w rozważaniach o sztucznej inteligencji', *Ethos* **69–70**, 239–251.

Łupkowski, P. [2006], 'Some Historical Remarks on Block's "Aunt Bubbles" Argument', *Mind and Machines* **16**(4), 437–441.

Łupkowski, P. [2009], The Question of OCR-based CAPTCHAs Safety, *w:* R. S. Choraś, A. Zabłudowski, (red.), 'Image Processing & Communications. Challenges', Academic Publishing House EXIT, Warszawa, s. 218–224.

Łupkowski, P., Urbański, M. [2008*a*], 'SemCAPTCHA — user-friendly alternative for OCR-based CAPTCHA systems', *Speech and Language Technology* **11**, 278–289.

Łupkowski, P., Urbański, M. [2008*b*], SemCAPTCHA. Telling Computers and Humans Apart by Means of Linguistic Competence and Positive Semantic Priming, *w:* 'Computational Intelligence: Methods and Applications', Academic Publishing House EXIT, s. 525–531.

Marciszewski, W. [1995], 'O przyszłości bez emocji', *Znak* **47**(484), 84.

Marciszewski, W. [1998], *Sztuczna inteligencja*, Krótko i węzłowato, Wydawnictwo Znak, Kraków.

Mauldin, M. L. [1994], Chatterbots, TinyMuds, and the Turing test: entering the Loebner Prize competition, *w:* 'Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-04)', American Association for Artificial Intelligence, Menlo Park, CA, USA, s. 16–21.

May, M. [2005], 'Inaccessibility of CAPTCHA. alternatives to visual turing tests on the web. w3c working group note 23 november 2005.'.
URL: *http://www.w3.org/TR/turingtest*

McCarthy, J., Hayes, P. J. [1969], Some philosophical problems from the standpoint of artificial intelligence, *w:* B. Meltzer, D. Michie, (red.), 'Machine Intelligence 4', Edinburgh University Press, s. 463–502.

McKinstry, C. [1997], 'Minimum Intelligence Signal Test: an Objective Turing Test', *Canadian Artificial Intelligence* s. 17–18.

McKinstry, C. [2009], Mind as Space: Toward the Automatic Discovery of a Universal Human Semantic-affective Hyperspace – A Possible Subcognitive Foundation of a Computer Program Able to Pass the Turing Test, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 17, s. 283–300.

Millar, P. H. [1973], 'On the point of the imitation game', *Mind* **LXXXII**(328), 595–597.

Moor, J. [1976], 'An Analysis of the Turing Test', *Philosophical Studies* **30**, 249–257. Przedruk w: S. Sheiber, red. [2004], The Turing Test..., s. 297–306.

Moor, J. [1978], ‘Explaining computer behavior’, *Philosophical Studies* **34**, 325–327. Przedruk w: S. Sheiber, red. [2004], The Turing Test..., s. 311–313.

Moor, J. [2001], ‘The Status and Future of the Turing Test’, *Mind and Machines* **11**, 79–93.

Mori, G., Malik, J. [2003], ‘Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA’.
URL: *http://www.cs.berkeley.edu/~mori/research/papers/*

Moy, G., Jones, N., Harkless, C., Potter, N. [2004], ‘Distortion estimation techniques in solving visual CAPTCHAS’, *IEEE CVPR* .

Münch, D. [1990], Mind, brains and cogntive science, *w:* A. Burkhardt, (red.), ‘Speech Acts, Meaning and Intentions. Critical Approaches to the Philosophy of John R. Searle’, De Gruyter, s. 367–390.

Murawski, R. [2000], *Funkcje rekurencyjne i elementy metamatematyki. Problemy zupełności, rozstrzygalności, twierdzenia Gödla*, Wydawnictwo Naukowe UAM, Poznań.

Naor, M. [1996], ‘Verification of a human in the loop or identification via the Turing test’.
URL: *http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.ps*

Naur, P. [1986], ‘Thinking and Turing’s test’, *BIT* **26**, 175–187.

Newman, A. H., Turing, A. M., Jefferson, G., Braithwaite, R. B. [1952], ‘Can automatic calculating machines be said to think?, broadcast discussion transmited on BBC (14 and 23 Jan. 1952)’, The Turing Digital Archive (www.turingarchive.org), Contents of AMT/B/6.

Nęcka, E., Orzechowski, J., Szymura, B., (red.) [2006], *Psychologia poznawcza*, Vol. 2, ACADEMICA & PWN, Warszawa.

Nęcka, E., (red.) [2005], *Inteligencja. Geneza – Struktura – Funkcje*, Postępy psychologii, GWP, Gdańsk.

Oakley, B., Gallehawk, J. [2007], Polski wkład w sukcesy GC&CS w Bletchley Park w czasie drugiej wojny światowej, *w:* S. Jakóbczyk, J. Stokłosa, (red.), ‘Złamanie szyfru Enigma. Poznański pomnik polskich kryptologów’, Wydawnictwo Poznańskiego Towarzystwa Przyjaciół Nauk, s. 69–105. Przeł. M. Grajek.

Papadimitriou, C. H. [2002], *Złożoność obliczeniowa*, Wydawnictwo Naukowo-Techniczne, Warszawa. Przeł. P. Kanarek i K. Loryś.

Peirce, Ch., S. [1931/1958], *Collected Works*, Harvard University Press, Cambridge, MA. (red.), Charles Hartshorne, Paul Weiss, Arthur W. Burks.

Pellen, L. [2009], How not to Imitate a Human Being: An Essay on Passing the Turing Test, *w:* G. B. R. Epstein, G. Roberts, (red.), ‘Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer’, Springer Publishing Company, Incorporated, rozdział 25, s. 431–446.

Penrose, R. [1995], *Nowy umysł cesarza. O komputerach, umyśle i prawach fizyki*, PWN, Warszawa. Przeł. P. Amsterdamski.

Penrose, R. [2000], *Cienie umysłu. Poszukiwanie naukowej teorii świadomości*, Zysk i S-ka, Poznań. Przeł. P. Amsterdamski.

Penrose, R. [2001], 'Kto skonstruuje myślącą isotę, będzie miał wszystkie prawa i obowiązki boga', *Wiedza i Życie* (9), 20–23.

Piccinini, G. [2000], 'Turing's rules for the imitation game', *Mind and Machines* **10**, 573–582.

Piccinini, G. [2003], 'Alan Turing and the Mathematical Objection', *Mind and Machines* **13**, 23–48.

Purtill, R. L. [1971], 'Beating the imitation game', *Mind* **LXXX**(318), 290–294. Przedruk w: S. Sheiber, red. [2004], The Turing Test..., s. 163–171.

Putnam, H. [1975], Robots: Machines or artificially created life?, *w:* H. Putnam, (red.), 'Mind, Language and Reality', Cambridge University Press, Cambridge, s. 386–407.

Reiter, R. [2001], *Knowledge in action: logical foundations for specifying and implementing dynamical systems*, MIT Press, Cambridge, Mass.

Ronald, E. M. A., Sipper, M. [2001], 'Intelligence is not enough: On the socialization of talking machines', *Mind and Machines* **11**, 567–576.

Rui, Y., Liu, Z. [2004], 'Artifacial: Automated reverse Turing test using FACIAL features', *Multimedia System* **9**, 493–502.

Sampson, G. [1973], 'In defence of Turing', *Mind* **LXXXII**(328), 529–594. Przedruk w: S. Sheiber, red. [2004], The Turing Test..., s. 173–175.

Sato, Y., Ikegami, T. [2004], 'Undecidability in the imitation game', *Mind and Machines* **14**, 133–143.

Saygin, A. P., Cicekli, I., Akman, V. [2001], 'Turing Test: 50 Years Later', *Mind and Machines* **10**, 463–518.

Schweizer, P. [1998], 'The truly total Turing test', *Mind and Machines* **8**, 236–272.

Searle, J. R. [1980], 'Minds, brains, and programs', *Behavioral and Brain Sciences* **3**(3), 417–457.

Searle, J. R. [1995], *Umysł, mózg i nauka*, PWN, Warszawa. Przeł. J. Bobryk.

Searle, J. R. [2009], The Turing Test: 55 Years Later, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 10, s. 139–150.

Shagrir, O. [2002], 'Effective computation by humans and machines', *Mind and Machines* **12**, 221–240.

Shanahan, M. [1999], Folk psychology and naïve physics, *w:* A. Clark, P. Millican, (red.), 'Connectionism, Concepts and Folk Psychology. The Legacy of Alan Turing, vol 2', Oxford University Press, s. 169–180.

Shanahan, M. [2003], The frame problem, *w:* E. N. Zalta, (red.), 'The Stanford Encyclopedia of Philosophy', Stanford University.
URL: *http://plato.stanford.edu/*

Shieber, S. M. [1994], 'Lessons from a restricted Turing test', *Communications of the ACM* **37**(6), 70–78.

Shieber, S. M. [2006], Does the Turing Test demonstrate intelligence or not?, *w:* 'Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)', s. 1539–1542.

Shieber, S. M. [2007], 'The Turing test as Interactive Proof', *Noûs* **4**(41), 686–713.

Shieber, S., (red.) [2004], *The Turing Test. Verbal Behavior as the Hallmark of Intelligence*, The MIT Press, Cambridge, Massachutetts, London.

Shoesmith, D. J., Smiley, T. J. [1978], *Multiple-conclusion Logic*, Cambridge University Press, Cambridge.

Stalker, D.F. [1976], 'Why machines can't think: A reply to James Moor', *Philosophical Studies* **34**, 317–320. Przedruk w: S. Sheiber, red. [2004], The Turing Test..., s. 307–310.

Stanford, T., Web, M. [2006], *100 sposobów na zgłębienie tajemnic umysłu*, Helion-O'Reilly, Gliwice. Przeł. E. Borówka i D. Kuczyńska-Szymała.

Sterrett, S. G. [2000], 'Turing's two tests for intelligence', *Mind and Machines* **10**, 541–559.

Straffin, P. D. [2001], *Teoria gier*, Scholar, Warszawa. Przeł. J. Haman.

Strelau, J., (red.) [1987], *O inteligencji człowieka*, Wiedza Powszechna, Warszawa.

Strelau, J., (red.) [2000], *Psychologia. Podręcznik akademicki*, Vol. 2, Gdańskie Wydawnictwo Psychologiczne, Gdańsk.

Szumakowicz, E. [2000], Sztuczna inteligencja — problem czy pseudoproblem, *w:* E. Szumakowicz, (red.), 'Granice sztucznej inteligencji, eseje i studia', Wydawnictwo Politechniki Krakowskiej, Kraków.

Tanimoto, S. L. [1987], *The Elements of Artificial Intelligence. An Introduction Using LISP*, Principles of Computer Science, Computer Science Press.

Thro, E. [1994], *Sztuczne życie — zestaw narzędzi badacza*, Intersoftland, Warszawa. Przeł. M. Syczewska.

Turing, A. M. [1939], 'Systems of logic based on ordinals', *Proceedings of the London Mathematical Society* **45**. Dostępne w The Turing Digital Archive (www.turingarchive.org), Contents of AMT/B/15.

Turing, A. M. [1948], 'Intelligent machinery', The Turing Digital Archive (www.turingarchive.org), Contents of AMT/C/11.

Turing, A. M. [1950], 'Computing machinery and intelligence', *Mind* **LIX**(236), 443–455. Polskie tłumaczenia: (1) Maszyny myślące a inteligencja, *w:* E. A. Feigenbaum, J. Feldman (red.), 'Maszyny matematyczne i myślenie', PWN, Warszawa 1972, s. 24–47. Przeł. D. Gajkowicz. (2) Maszyny myślące a inteligencja, *w:* B. Chewdeńczuk (red.), 'Filozofia umysłu', Wydawnictwo Spacja, Warszawa 1995, s. 271–300. Przeł. M. Szczubiałka.

Turing, A. M. [1951*a*], 'Can digital computers think?', The Turing Digital Archive (www.turingarchive.org), Contents of AMT/B/5.

Turing, A. M. [1951*b*], 'Intelligent machinery, a heretical theory', The Turing Digital Archive (www.turingarchive.org), Contents of AMT/B/4.

Turing, A. M. [1952], 'The chemical basis of morphogenesis', *Philosophical Transactions of the Royal Society of London* **237**(641). Dostępne w The Turing Digital Archive (www.turingarchive.org), Contents of AMT/B/22.

Turney, D. T. [2001], 'Answering subcognitive Turing test questions: A reply to French', *Journal of Experimental and Theoretical Artificial Intelligence* **13**(4), 409–419.

Urbański, M. [2001], 'Synthetic tableaux and erotetic search scenarios: Extension and extraction', *Logique & Analyse* **173–174–175**, 69–91.

Urbański, M. [2005], O rozumowaniach abdukcyjnych, *w:* T. Mróz, M. Sieńko, (red.), 'Propositiones', Instytut Filozofii Uniwersytetu Zielonogórskiego, s. 143–150.

Urbański, M., Wiśniewski, A. [2006], Socratic trees. Manuskrypt.

van Vugt, H. C., Konijn, E. A., Hoorn, J. F., Keur, I., Eliëns, A. [2007], 'Realism is not all! User engagement with task-related interface characters', *Interacting with Computers* **19**(2), 267–280.

von Ahn, L., Maurer, B., Mcmillen, C., Abraham, D., Blum, M. [2008], 'reCAPTCHA: Human-Based Character Recognition via Web Security Measures', *Science* **321**, 1465–1468.

Vora, P. [2009], *Web Application Design Patterns*, Morgan Kaufmann.

Watt, S. [1996], 'Naive psychology and the inverted Turing test', *Psycholoquy* **7**(14).
URL: *http://www.cogsci.esc.soton.ac.uk/cgi/psyc/newpsy?7.17*

Watt, S. [2009], Can people think? or machines? a unified protocol for turing testing, *w:* G. B. R. Epstein, G. Roberts, (red.), 'Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer', Springer Publishing Company, Incorporated, rozdział 18, s. 301–318.

Whitby, B. [1996], The Turing Test: AI's biggest blind alley?, *w:* P. Millican, A. Clark, (red.), 'Machines and thought', Vol. 1, Clarendon, Oxford, rozdział 3.

Whitby, B. [1997], 'Why the turing test is AI's biggest blind alley'.
URL: *http://www.cogs.susx.ac.uk/users/blayw/tt.html*

Wiśniewski, A. [1990], *Stawianie pytań — logika i racjonalność*, Wydawnictwo UMCS, Lublin.

Wiśniewski, A. [1995], *The Posing of Questions: Logical Foundations of Erotetic Inferences*, Kluwer AP, Dordrecht, Boston, London.

Wiśniewski, A. [2001], 'Questions and inferences', *Logique & Analyse* **173-175**, 5–43.

Wiśniewski, A. [2003], 'Erotetic search scenarios', *Synthese* **134**, 389–427.

Wiśniewski, A. [2004], 'Erotetic search scenarios, problem-solving, and deduction', *Logique & Analyse* **185-188**, 139–166.

Wiśniewski, A. [2008], 'Questions, inferences, and dialogues'. Prezentacja przedstawiona na konferencji LONDIAL2008.

Wiśniewski, A., Pogonowski, J. [2010], 'Interrogatives, Recursion, and Incompleteness', *Journal of Logic and Computation* **20**(6), 1187–1199.

Woleński, J. [1999], *Metamatematyka a epistemologia*, PWN, Warszawa.

Yan, J., El Ahmad, A. S. [2007], Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorighms, *w:* 'Proc. of the 23rd Annual Computer Security Appliccations Conference (ACSAC'07)Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorighms', s. 279–291.

Yan, J., El Ahmad, A. S. [2008*a*], A Low-cost Attack on a Microsoft CAPTCHA, Technical report, School of Computing Science Technical Report. Newcastle University, England.

Yan, J., El Ahmad, A. S. [2008*b*], Usability of CAPTCHAs. Or usability issues in CAPTCHA design. Symposium On Usable Privacy and Security (SOUPS). July 23-25 2008, Pittsburgh, PA, USA.

Yan, J., El Ahmad, A. S. [2009], 'CAPTCHA Security: A Case Study', *IEEE Security and Privacy* **7**(4), 22–28.

Zdenek, S. [2001], 'Passing Loebner's Turing Test: A Case of Conflicting Discourse Functions', *Mind and Machines* **11**, 53–76.

Indeks

ISBN 978-83-232-2208-8